

# GRÁFICOS NO TK 90X



Victor Mirshawka

Victor Mirshawka

GRÁFICOS NO TK 90X

nobel

Neste livro introduzem-se crianças e adultos para o mundo fascinante da ciência, ilustrada pelos gráficos, através da tela conectada a esta maravilha eletrônica que é o microcomputador TK 90X.

O livro, nos seus 20 capítulos apresenta diversos programas em BASIC que permitem a você obter espirais, circuitos eletrônicos, visão tridimensional dos corpos, o seu biorritmo um simulador lógico etc.

Toda a potencialidade de seu TK 90X, principalmente no que se refere à parte gráfica, é ilustrada com estes programas.

Você sabia que, entre outras coisas, o TK 90X permite criar os seus próprios caracteres gráficos?

As possibilidades, as descobertas e a alegria são sem limites depois da exibição na tela dos resultados de todos os programas apresentados.

Tudo o que você deve fazer para atingir este "estado de graça" é ter estes livros e o TK 90X, sentar à sua frente e, com o livro ao lado, começar a teclar desesperadamente...

ISBN 85-213-0326-2



nobel

**GRÁFICOS**  
*NO*  
**TK90X**

M655g Mirshawka, Vitor, 1941-  
Gráficos no TK 90X / Victor Mirshawka. — São  
Paulo : Nobel, 1985.

ISBN 85-213-0325-4

1. BASIC (Linguagem de programação para computadores) 2. Gráficos por computador 3. TK 90X (Computador) — Programação I. Título.

85-1244

17. CDD-651.8  
18. -001.642  
17. -001.6424  
18. -001.6425

Índices para catálogo sistemático:

1. BASIC : Línguagem de programação : Computadores : Processamento de dados 651.8 (17.) 001.6424 (18.)
2. Gráficos : Computadores : Programas : Processamento de dados 651.3 (17.) 001.6425 (18.)
3. TK 90X : Computadores : Programação : Processamento de dados 651.8 (17.) 001.642 (18.)

*Victor Mirshawka*

Professor titular de Cálculo Numérico, Estatística e Pesquisa Operacional da Faculdade de Engenharia da Fundação Armando Álvares Penteado.

Professor titular de Cálculo Numérico e Ciência da Computação, Estatística e Pesquisa Operacional da Universidade Mackenzie.

Professor associado do departamento fundamental da Escola de Engenharia Mauá.

Mestre em Estatística pela Universidade de São Paulo.

# GRÁFICOS NO TK90X

1<sup>a</sup> edição  
1<sup>a</sup> reimpressão  
1986

CAPA E ILUSTRAÇÕES: Lirio Fissac Yuasa

Nobel

## APRESENTAÇÃO

Livraria Nobel S.A.  
Rua da Balsa, 559  
02910 – São Paulo – SP

### É PROIBIDA A REPRODUÇÃO

Nenhuma parte desta obra poderá ser reproduzida sem a permissão por escrito dos editores através de qualquer meio: Xerox, fotocópia, fotográfico, fotomecânico. Tampouco poderá ser copiada ou transcrita, nem mesmo transmitida através de meios eletrônicos ou gravações. Os infratores serão punidos através da Lei 5.988, de 14 de dezembro de 1973, artigos 122-130.

Neste livro introduzem-se crianças e adultos para o mundo fascinante da ciência, ilustrada pelos gráficos, através da tela conectada e esta maravilha eletrônica que é o microcomputador TK 90X.

O livro, nos seus 20 capítulos contém muitos programas (31 no total) todos apresentados na linguagem BASIC para o TK 90X.

Estes programas "invadem" as mais diversas áreas como a Matemática, a Física, a Eletrônica, a Química, o Desenho, etc.

Toda a potencialidade do seu TK 90X, principalmente no que se refere à parte gráfica é ilustrada com estes programas.

Você verá lindas curvas (senóides, círculas, elipses), poderá saber como é o seu bioritmo, aproximará ou afastará de si lindo octaedro, brincará com todas as opções permitidas para as portas lógicas, viverá a visão tridimensional com o chapéu mexicano, etc.

As possibilidades, as descobertas e a alegria são sem limites depois que você estimado(a) leitor(leitora) ver exibidos na tela os resultados de todos os programas do livro.

E útil ressaltar que neste livro não existe a preocupação de fornecer explicação detalhada para as instruções que constituem os programas.

Caso o(a) respeitado(a) leitor(leitora) tenha dúvida sobre o desempenho de alguns comandos deve recorrer ao livro "BASIC no TK 90X" que é também de minha autoria.

Porém, a partir deste momento, tudo o que você deve fazer para atingir um "estado de graça" é com este livro ao lado, sentar em frente ao seu TK 90X e começar a teclar.

## AGRADECIMENTOS

Foi contínua e cada vez mais indispensável a ajuda da minha querida esposa e amiga Nilza Maria e dos meus três filhos Victor Junior, Sérgio e Alexandre (Sacha para os íntimos, que agora já ajudou...).

Eu sei que para aqueles que me prestigiam usando os meus livros de Informática, a primeira parte da frase acima parece um "repeteco" um tanto sonso...

Pois então vou melhorar o meu depoimento dizendo que Nilza Maria já me ajudava com os meus livros na minha primavera (escrevi o meu primeiro livro em 1958) e no verão da minha existência colabora revendo, corrigindo e indicando a melhor forma de apresentar um assunto.

Estou a um passo de entrar no outono da minha vida e com que alegria vejo o primeiro livro publicado pelos meus filhos Victor Junior e Sérgio.

Estamos certos Nilza Maria e eu, que se Deus nos permitir chegar ao inverno da existência ele não será frio, mas muito ao contrário acalentador e repousante por termos deixado um bom exemplo, pelo menos para os nossos filhos.

Muito obrigado ao amigo George Kovari, diretor da Microdigital, por mais uma vez ter me engajado neste projeto muito mais educacional do que comercial.

Estou seguro que o TK 9 $\varnothing$ X irá modificar de forma substancial o ensino, dinamizando-o e principalmente colocando a disposição dos que estudam e trabalham os mais fantásticos recursos para texto, cálculos, gráficos e som.

"Arigatô", Toshiro Iqueda, datilógrafo, editor e amigo de muitos anos (datilografou todos os meus livros a partir do terceiro e hoje já tenho mais de cinquenta títulos publicados...).

Finalmente e com muita sinceridade sou grato mais uma vez aos amigos da Livraria Nobel S.A., Ary K. Benclowicz e Luigi Zamboni por acreditarem em tudo o que escrevo e por investirem nisto de corpo e alma colocando toda a empresa a minha disposição.

Assim até parece fácil escrever e editar um livro...

## SUMÁRIO

Capítulo 1	Execução de figuras .....	1
Capítulo 2	Gráficos trigonométricos .....	5
Capítulo 3	Três em um .....	8
Capítulo 4	Representação ampliada .....	14
Capítulo 5	Elaborando uma histograma gaussiano e variantes .....	16
Capítulo 6	Fazendo esboço do teclado do TK-9 $\varnothing$ X .....	21
Capítulo 7	Colocando coisas em destaque .....	24
Capítulo 8	Balança .....	28
Capítulo 9	Química .....	33
Capítulo 10	Probabilidades .....	37
Capítulo 11	Fahrenheit x Celsius .....	41
Capítulo 12	Aeroplano .....	44
Capítulo 13	Lote econômico .....	47
Capítulo 14	Chapeu mexicano .....	50
Capítulo 15	Bioritmo .....	55

Capítulo 16	
Representação de uma função .....	61
Capítulo 17	
Espirais .....	68
Capítulo 18	
Perspectiva cônica .....	76
Capítulo 19	
Uma aplicação para a Eletrônica com direito a uma distração - tipo destreza no circuito.	84
Capítulo 20	
Simulador lógico além de um super jogo .....	96

# Capítulo 1

## Execução de figuras

Os pequenos programas que se seguem vão lhe dar os primeiros indícios das potencialidades do seu TK 90X na geração de figuras e curvas.

O primeiro programa desenha uma série de curvas senoidais.

```

10 LET r=RND*6 : PAPER r : BORDER r : INK 7
15 CLS
20 FOR j=1 TO 2
30 FOR a = 0 TO 50 STEP 10
40 FOR x = 0 TO 3 STEP 0.05
50 IF j=1 THEN PLOT 20*x + a, 38*SINx + 38 + a
60 IF j=2 THEN PLOT OVER 1; 20*x + a, 38*SINx + 38 + a
70 NEXT x
80 NEXT a
90 NEXT j
100 CLS

```

Execute e veja se sai algo do seguinte tipo:

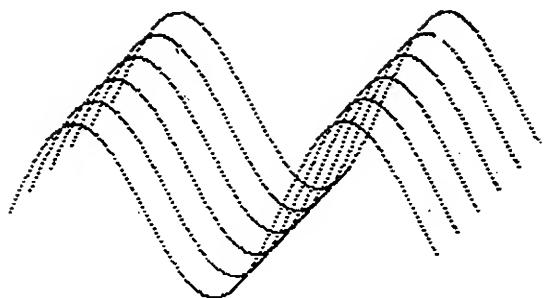


Figura 1a

Observação Importante (O.I.)

Tente entrar no lugar do  $\sin x$  com  $\cos x$ , mude os passos nas linhas 20 e 30, introduza outros valores na linha 50 (por exemplo 25 no lugar de 20 e 30 no lugar de 38) etc..

Mexa-se e obtenha outras lindas figuras com o programa.

O programa que vem a seguir representa o gráfico de uma função depois que você introduzir a mesma dentro do TK 90X.

Aí vêm algumas sugestões para a  $f(x)$

$$\frac{1}{3}x+4, x^2, \frac{10000}{300-x}, e^x, \sqrt{x}, \text{ etc.}$$

```

10 INPUT INK 3; "Escreva uma função de x"; LINE f$
20 LET r = RND*5: PAPER r: BORDER r: INK 6: CLS
30 PLOT 0,175: DRAW 0,-175
40 DRAW 255,0
50 FOR x = 0 TO 255
60 LET y = VAL f$
70 IF x > 255 OR y > 175 THEN GO TO 110
80 PLOT x,y
90 SOUND 0.018,12
100 NEXT x
110 PRINT AT 13,2; FLASH 1; "y="; f$
```

O.I. - Você caro(a) leitor (leitora) tem um programa bem mais sofisticado que este com a mesma finalidade com o nome "REPRESENTAÇÃO DE UMA FUNÇÃO" - Capítulo 16.

O último mini-programa do tópico "Execução de figuras" desenha circunferências com centros diferentes e com diferentes raios (menor e maior) com o que se obtém circunferências "sobrepostas".

```

10 REM **Desenho de muitas circunferências
20 INPUT "Entre com a abscissa (x)"; x
30 INPUT "Entre com a ordenada (y)"; y
40 INPUT "Qual é o raio menor (1-25)?"; rmin
50 INPUT "Qual é o raio maior (20-50)?"; rmax
60 INPUT "Qual é a cor?"; c
70 FOR z = rmin TO rmax
80 CIRCLE INK c; x,y,z
90 NEXT z
100 GOTO 20
```



Execute e veja se sai algo do tipo:



Figura 1b



O.I. - Toda vez que na listagem de um programa aparecer o caractere "□", isto significa que é para deixar um espaço em branco.

## Capítulo 2

# Gráficos trigonométricos

Inicialmente vejamos como se pode obter um conjunto de padrões gráficos descobertos no século XIX pelo físico francês Jules Lissajous que estudou a composição dos movimentos vibratórios mediante um processo ótico.

Hoje é muito simples obter a figura de Lissajous indicada na Figura 2.

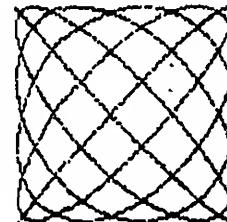


Figura 2 -  $p = 1.1$

Aí vai o programa que lhe permite obter isto:

```

10 INPUT "Entre com o parâmetro (0.7 a 1.3)"; p: CLS
20 BORDER 1: INK 2: PAPER 6
30 LET t = 0
40 LET t = t+.1
50 LET x = 50 * (1+SIN(p*t))
60 LET y = 50 * (1+COS t)
70 PLOT x + 50, y + 50
80 GOTO 40

```



Observação importante O.I.

Qualquer dúvida que você tiver sobre as instruções deve recorrer ao livro "BASIC NO TK 90X" de nossa autoria obviamente...

Uma segunda ilustração no seu TK 90X daqui lo que já lhe deu (ou lhe dará) muito trabalho é o gráfico da função  $\frac{\sin x}{x}$ .

Aqui vai a simples e musical representação da própria

```

10 CLS
20 INK 2: PAPER 6: BORDER 1
30 FOR j=1 TO 250
40 IF j=125 THEN GOTO 60
50 LET x = (j-125)/5
60 LET y = 80 * SIN(x)/x
70 PLOT j,y: DRAW 1,y
80 SOUND .018, y/2
90 NEXT j
100 GOTO 10

```



Para finalizar este pequeno compacto do uso das funções seno e cosseno aí vai o programa que dá o gráfico da rosácea de quatro folhas cuja equação polar é:

$$r = a \cos 2t \quad (0 \leq t \leq 2\pi, \quad a = 50)$$

usando-se as expressões paramétricas

$$\begin{cases} x = r \sin t \\ y = r \cos t \end{cases}$$

```

10 INK 2: PAPER 7: BORDER 1
20 LET I=0: LET F=2*PI
30 LET N=90
40 LET A=50
50 LET INC = (F-I)/N
60 FOR j=I TO F STEP INC
70 LET R=A*COS(2*j)
80 LET X=R*SIN j
90 LET Y=R*COS j
100 PLOT 127+x, 87+y
110 NEXT j
120 STOP

```



# Capítulo 3

## Três em um

Como existe apenas um número finito de pixels disponível na tela, as linhas retas desenhadas na diagonal parecem muitas vezes com "escadinhas".

Na prática isto não representa nenhum problema real e até pode ser convertido em uma vantagem como é o caso do primeiro programa deste capítulo no qual se desenha uma sequência de 90 belos quadros.

Este programa, por incrível que possa parecer utiliza os pequenos defeitos dos aparelhos de T.V. que tendem a não apresentar os detalhes mais finos.

Bem chega de papo, é você mesmo que vai notar e julgar a beleza e o aspecto artístico provocado pelo programa.

```

10 FOR q=1 TO 90
20 LET p=2 + INT(3*RND)
30 LET c=INT(2*RND)
40 FOR x=255 TO 0 STEP -p
50 PLOT x,0: DRAW OVER c; 255 -2*x, 175
60 NEXT x
70 FOR y=0 TO 175 STEP p
80 PLOT 0,y: DRAW OVER c; 255, 175 - 2*y
90 NEXT y
100 NEXT q

```

O.I. Aí vêm algumas rápidas explicações

- 1) Da linha 10 até a linha 100 tem-se o LOOP que serve para imprimir os 90 quadros.
- 2) Na linha 20 tem-se a instrução que faz a escolha aleatória do passo para as linhas 40 e 70 sendo que o valor do passo está sempre entre 2 e 4 (extremos incluídos).
- 3) Na linha 30 tem-se a escolha aleatória do valor de c que pode ser 0 ou 1.  
Este código é usado nas linhas 50 e 80 após a instrução DRAW OVER para fazer com que o programa faça o TK 90X desenhar linhas pretas (quando c=0) ou desenhar linhas invertendo o que já está na tela (c=1).
- 4) O LOOP que vai das linhas 40 até 60 desenha uma metade do quadro e o que vai das linhas 70 até 90 a outra metade.

Execute o programa e veja se obtém algo parecido com o que está na Figura 3.



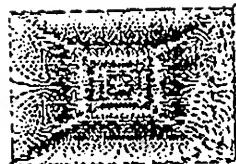


Figura 3

Você percebeu o lindo efeito provocado pela instrução DRAW.

Porém ela tem muito mais coisa embutida dentro das suas possibilidades.

Ao se acrescentar um terceiro valor após a instrução DRAW ou seja escrevendo-se

DRAW x,y,a

o TK 9@X desenharia um arco de circunferência.

Para ter uma imediata demonstração deste efeito tecle o seguinte programinha:

```
10 FOR p = -1.5*PI TO 1.5*PI STEP PI/2
20 PLOT 127,60
30 DRAW 0,60,p
40 NEXT p
```

Execute e veja se obtém o que está indicado na Figura 4.

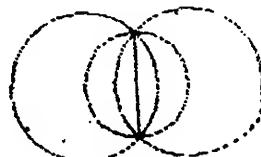


Figura 4

Você vai sentir nos diversos programas deste livro que a instrução DRAW é usada como parte de uma rotina para desenhar as mais complicadas formas geométricas mas neste momento quero lhe dar um pouco de distração e por isto aí vai um programa no qual se apresenta um joguinho baseado muito no poder da instrução DRAW.

A finalidade do jogo é você "capturar" o número máximo possível de cápsulas espaciais escolhendo um campo de força adequado em torno delas.

O campo de força tem a forma de uma bolha e você pode controlar o seu tamanho,

Existe apenas um pequeno problema ou seja se você tocar com o contorno da bolha em alguma das cápsulas espaciais desativará o mesmo por ter provocado um curto-circuito.

Neste jogo podem participar dois ou mais contendores cada um deles sendo eliminado do jogo quando ocorrer a destruição do seu campo de força.

Aí vai o programa:

```
5 INK 3: BORDER 4: PAPER 6: CLS
10 FOR j=1 TO 17: PRINT AT 15*RND, 30*RND; "0": NEXT j
20 PLOT 80,10: DRAW 0, -10: DRAW 85,0: DRAW 0,10
30 INPUT "Entre com o tamanho da bolha (1-9)": tamanho
40 IF tamanho < 1 OR tamanho > 9 THEN GOTO 30
50 PLOT 80,10: DRAW OVER 1; 85,0,-(4.35+tamanho/10)
60 PRINT AT 19,5; Vai tentar outra vez? (S/N)"
70 IF INKEY$ = "s" OR INKEY$ = "S" THEN RUN
80 IF INKEY$ <> "n" OR INKEY$ <> "N" THEN GOTO 70
```

Algumas explicações sobre este programa

- 1) Linha 10 - Imprimem-se com esta linha de instruções múltiplas as 17 cápsulas espaciais em posições aleatoriamente escolhidas.

- 2) Linha 2 $\emptyset$  - Aí estão as instruções que "forçam" o TK 9 $\emptyset$ X a desenhar o emissor do campo de força.  
 3) Linhas 3 $\emptyset$  a 5 $\emptyset$  - Aí é feita a escolha (por você) do tamanho da bolha, verifica-se a sua validade (linha 4 $\emptyset$ ) e então desenha a "bolha".

A função OVER 1 é usada na linha 5 $\emptyset$  para mostrar que há destruição do campo de força quando ele toca uma cápsula espacial.

- 4) Linhas 6 $\emptyset$  a 8 $\emptyset$  - Neste trecho espera-se pela entrada do próximo jogador na partida.

Ao executar o programa deve surgir algo do tipo mostrado na Figura 5.

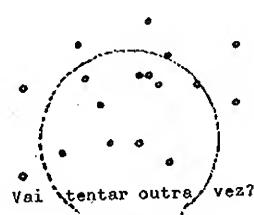


Figura 5

Para finalizar o Capítulo "TRÊS EM UM" vamos desenhar algumas quadricircunferências ou alguns retículo-círculos.

Cabe a você ficar com o nome que gostar mais.

Você já notou que a instrução DRAW permite desenhar um trecho ou seja um arco de circunferência e faz isto calculando as coordenadas dos vários pontos do arco e então unindo os pontos por linhas retas.

Comumente não se comenta isto, mas quando você tornar o terceiro parâmetro após a instrução DRAW extremamente grande o resultado pode ser tremendamente inesperado e espetacular.

Entre com o seguinte programa e atribua ao parâmetro p valores altos.

1 $\emptyset$  INPUT p: PRINT p  
 2 $\emptyset$  PLOT 127,1 $\emptyset$ : DRAW  $\emptyset$ ,15 $\emptyset$ ,p\*PI

ou melhor, se você quiser ver desenhos como os indicados nas Figura 6a),b),c) e d) no lugar da linha 2 $\emptyset$  entre com a nova linha 2 $\emptyset$

2 $\emptyset$  PLOT 127,1 $\emptyset$ : DRAW OVER 1;  $\emptyset$ , 15 $\emptyset$ , p\*PI

Entre os valores "interessantes" que você deve testar indico

p = 101, 119, 253, 261, 271, 383 e 9999.

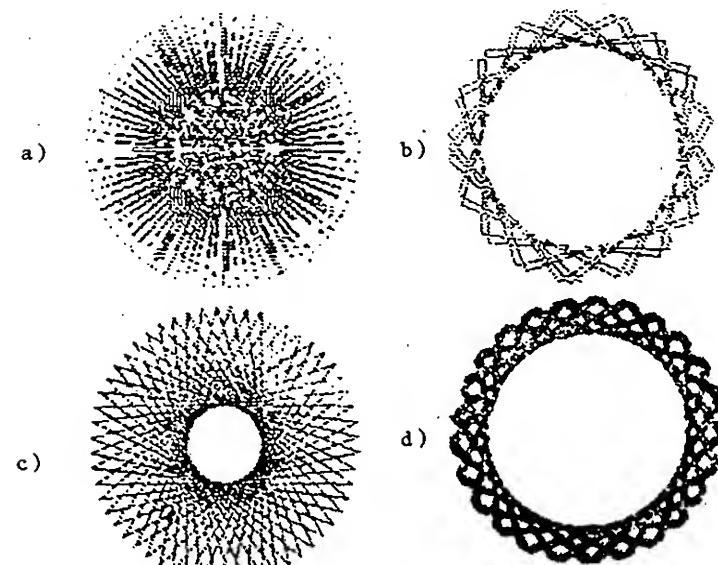


Figura 6



# Capítulo 4

## Representação ampliada

Aí vai um programa bem simplificado que lhe permite obter uma versão ampliada de qualquer caractere: normal, gráfico ou definido por você estimado(a) usuário(a).

É óbvio que não se permite o uso das palavras-chave.

Só é possível dispor de quatro caracteres por linha, podendo-se escolher a cor deles (número de 1 a 7).

É na variável e que se armazena o endereço de início, na ROM, do conjunto de caracteres.

```

10 REM *** REPRESENTAÇÃO AMPLIADA ***
20 SOUND .1,13: SOUND .1,13: SOUND .7,13: SOUND .8,13:
      SOUND .7,14
30 LET e = PEEK 23606 + 256 * PEEK 23607
40 INPUT "Entre com a cor da tinta □"; t: INK t
50 INPUT "Escreva uma palavra (não mais de 4 letras)"; p$
```

```

60 FOR v = 0 TO 3
70 LET f = e + CODE p$ (v+1)*8
80 FOR g = 0 TO 7
90 LET h = PEEK (f+g)
100 FOR j = 0 TO 7
110 IF h/2 <> INT(h/2) THEN PRINT AT 13+g, 8-j+(v*8);
      INK t;"█"
115 SOUND .008,t
120 LET h = INT (h/2)
130 NEXT j: NEXT g: NEXT v
140 INPUT "Quer escrever algo mais?□"; r$
150 POKE 23692,-1
160 IF r$(1) = "S" OR r$(1) = "s" THEN PRINT".....";
      GOTO 10
170 STOP
```

Execute e veja se sai algo deste tipo, quando na linha 110 você tiver o caractere gráfico "█"

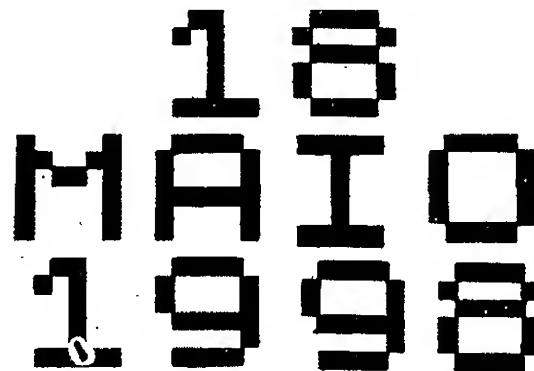


Figura 7

O.I. Note que cada linha deve receber obrigatoriamente quatro caracteres e caso envie menos complete a mesma com espaços em branco.

# Capítulo 5

## Elaborando um histograma gaussiano e variantes

Neste programa é muito importante o desempenho da função RND para produzir o histograma.

Aí está o programa

```

10 DIM a(16)
20 LET esc = .1
30 RAND
40 SOUND .1,12: SOUND .2,13:
      SOUND .3,14
50 LET c = 0
60 FOR i = 1 TO 12
70 LET c = c+RND
80 NEXT i
90 LET dp = 2: LET media = 7.5
100 LET j=INT(1+(c-6)*dp+média)

```



A variável "esc" é utilizada como um fator de escala. Quanto menor for o seu valor, mais números aleatórios poderão ser acumulados antes que o programa pare, porque uma das barras ou hastas atingiu o topo da tela. Sugiro (e não é nome de japonês) que você estimando(a) leitor(leitora) tente outros valores e sinta ou melhor veja,a diferença

```

110 IF j < 1 THEN LET j=1
120 IF j > 16 THEN LET j=16
130 LET a(j) = a(j) + esc
140 IF a(j) > 21 THEN STOP
150 PRINT AT 21-a(j), 2*j-1;"■"
160 GOTO 50

```

Estas instruções impedem que o programa se "auto destrua" caso sejam gerados valores muito extremos

Execute e veja se sai algo do tipo:

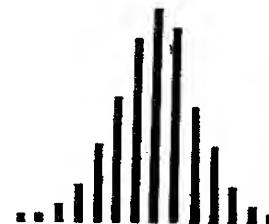


Figura 8

Caso você queira ir anotando como vai evoluindo o seu crescimento ou da altura de uma planta, como vai decrescer o poder aquisitivo do seu salário com o tempo, quantos são os acertadores da loteria esportiva em cada semana, etc. aí está o programa que lhe convém...

O TK 90X desenhárá os seus diagramas em seis cores distintas de forma que pode montar uma cor sobre a outra para mostrar de quanto foi a variação desde a última vez que ocorreu a comprovação.

O TK 90X fará com que se mostre na tela uma linha vertical (perpendicular) e uma linha de base (horizontal).

Além disto o TK 90X lhe perguntará em primeiro lugar os valores que desejas utilizar e a seguir lhe pedirá para que escolha a cor (1 a 6) na qual queres que seja impresso o diagrama.

Finalmente lhe pedirá que tecles os dados que desejas que apareçam representados no diagrama.

Por exemplo se você quer representar a variação da temperatura máxima de alguns dias consecutivos e se no primeiro dia a temperatura for de 35°C deves entrar com 35, se no dia seguinte a temperatura máxima for 28°C deves entrar com 28 etc.

Se em algum instante você desejar sair do programa tecle uma letra e o programa voltará a lista de cores.

Podes voltar a escolher outro valor para que o mesmo contraste com a cor escolhida anteriormente de forma que possa ver por exemplo a comparação das temperaturas das semanas anteriores ou até do mesmo mês, porém no ano anterior.

Note que as cores estão impressas acima das teclas dos números no seu TK 90X.

Quando quiseres parar com os dados basta entrar com uma STRING (linha 310).

Para parar totalmente com os dados basta teclar "n" ou "N" na linha 420.

Aí vai o programa

```

10 REM ** Histograma **
20 FOR j=0 TO 15
30 READ d
40 POKE USR "a "+j, d
50 NEXT j
60 DATA 127, 127, 127, 127, 127, 127, 127, 127
70 DATA 254, 254, 254, 254, 254, 254, 254, 254

```



```

80 INPUT "Qual é o seu valor máximo □ (múltiplos de 20)?"; m
90 IF m<20 THEN PRINT PAPER 4; INK 0; AT 10,9;""
Muito pequeno!": GOTO 80
100 IF m>100 THEN PRINT PAPER 3; INK 0; AT 10,9;""
Muito alto! □□□□": GOTO 80
110 INPUT "Qual é a cor (1-6)? □"; c
120 IF c<1 OR c>6 THEN GOTO 110
130 CLS
140 LET s = INT(m/20)
150 LET p=1
160 FOR k=1 TO 19
170 IF s*k>9 THEN LET p=0
180 PRINT AT 20-k, p; s*k
190 IF k=9 THEN LET p=20
200 NEXT k
210 LET coluna = 1
220 FOR i=1 TO 14
230 LET coluna = coluna+2
240 PRINT AT 21, coluna; i
250 NEXT i
260 PLOT 18,17
270 DRAW 0,-155: DRAW 230,0
280 SOUND .1,11: SOUND .2,12: SOUND .3,13
290 LET coluna = 3
300 INPUT "Qual é o valor? □"; v$
310 IF v$(1)>"9" OR v$(1)<"0" THEN GOTO 410
320 LET valor = VAL v$
330 IF valor >m THEN LET valor = m
340 FOR j=0 TO valor / s
350 IF j=0 THEN GOTO 370
360 PRINT INK c; AT 20-j, coluna; "AA"
370 NEXT j
380 LET coluna = coluna+2
390 IF coluna = 31 THEN GOTO 410

```

entre duas vezes  
com A em modo gráfico

```

400 GOTO 300
410 INPUT "Vai entrar com algo mais?"; r$
420 IF r$(1) = "N" OR r$(1) = "n" THEN STOP
430 INPUT "Qual é a cor agora?"; c
440 IF c < 1 OR c > 6 THEN GOTO 430
450 GOTO 290

```

Entre com os seus dados e veja se obtém algo parecido com o que se mostra na Figura 9.

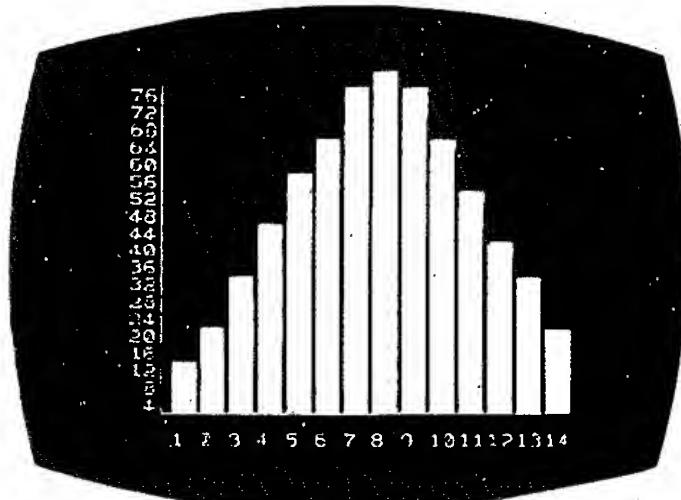


Figura 9

## Capítulo 6

### Fazendo esboço do teclado do ZX90X

Este programa lhe permite desenhar figuras na tela do aparelho de T.V.

Ele é muito instrutivo pois lhe permite sentir todo o poder da instrução PLOT e lhe dará muita experiência com o sistema de coordenadas (X, Y).

Não se deve deixar de citar o lindo efeito do INVERSE usado com a instrução PLOT.

O quadro ou a figura é desenhada utilizando-se as teclas 5,6,7 e 8 para mover um pequeno cursor "faiscante" ao redor da tela.

Como um auxílio (fundamental aliás...) caso você esteja procurando um desenho de muita precisão exibe-se no canto baixo da esquerda e isto continuamente a posição atual do cursor, isto é, as coordenadas do mesmo.

O cursor deixa uma linha preta atrás de si a medida que se desloca, a menos que você mantenha

a tecla SHIFT abaixada, quando então ele desenharia uma linha branca, permitindo-lhe apagar desta forma qualquer erro que tenha cometido.

Caso você tenha parado o programa com um **SHIFT-BREAK** e notar que algo mais poderia ser agregado a sua obra prima não entre com RUN pois assim você vai limpar tudo da tela.

Em vez disto entre com o comando **CONT (contine)**.

Finalmente o programa neste caso foi estruturado como se fosse um LOOP sem fim pois o que se deseja é uma atividade contínua e a cada passo mais criativa e imaginativa da sua parte.

Seria um tanto lusitano (sem mexer demais com os queridos descobridores...) a todo momento o programa lhe perguntar:

Você deseja continuar (S/N)?

```

10 LET x = 127: LET y = 88
20 FOR c = 0 TO 1: PLOT INVERSE c; x,y
30 LET k = CODE INKEY$ _____
40 IF k >=53 AND k <=56 THEN PLOT x,y
50 IF k >=8 AND k <=11 THEN PLOT INVERSE 1;x,y
60 PRINT AT 20,0; "X="; x; "□□□"; "y="; y; "□□□"
70 IF (k=8 OR k=53) AND x > 0 THEN LET x=x-1
80 IF (k=9 OR k=56) AND x < 255 THEN LET x=x+1
90 IF (k=10 OR k=54) AND y > 0 THEN LET y=y-1
100 IF (k=11 OR k=55) AND y < 175 THEN LET y=y+1
110 NEXT c
120 GOTO 20

```

Você já sabe, pois tem o meu livro o "BASIC no TK 99X" que a função ou variável INKEY\$ retorna um valor STRING. Como precisamos de um valor numérico é obrigatório o uso da instrução CODE para podermos lidar com o movimento do cursor sem impressão (códigos 8,9,10 e 11).



### Explicações adicionais

Linha 10 → Aí estão os valores iniciais para a abscissa(x) e para a ordenada (y)

Linha 20 a 110 → LOOP principal para ativar e apagar o cursor "piscante"

Linha 30 → O seu TK 99X espera que você aperte alguma tecla

Linha 40 → Se uma das teclas 5,6,7 ou 8 for apertada ter-se-á marcada a posição atual do cursor em preto

Linha 50 → Se a tecla SHIFT for pressionada juntamente com uma das teclas 5,6,7 ou 8 então ter-se-á marcada a posição atual do cursor em branco

Linhas 55 e 60 → Imprime-se o valor atual das coordenadas do cursor.

A linha 55 é necessária para que na próxima vez seja limpo o espaço ocupado pelos números que representavam as coordenadas da última posição

Linhas 70 a 100 → Aí são calculadas as novas coordenadas

Linha 120 → Volta-se para um novo "faiscamento" do cursor

Caso você esteja sem imaginação do que deve desenhar aí vai uma sugestão.

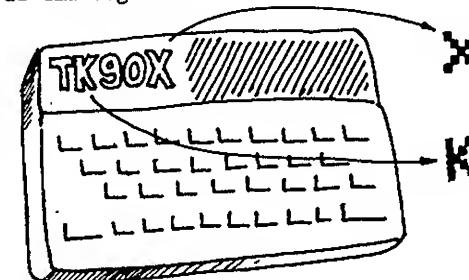


Figura 10

# Capítulo 7

## Colocando coisas em destaque

Este é um programa muito útil para apresentar mensagens de divulgação com o auxílio do caractere gráfico.

Inicialmente o programa permite criar uma tela em alguma cor escolhida e com as teclas "5", "6", "7" ou "8" pode-se controlar as direções.

A tecla "1" ordena que se desenhe e com a tecla "2" se manda imprimir um espaço em branco ou seja interrompe-se a impressão.

Depois que tiver desenhado tudo o que quiser na tela tecle um "c" de chega (por enquanto, pelo menos...) e aí sim escreva a mensagem que quer divulgar, não esquecendo de escolher a linha onde quer que a mesma apareça.

Depois de ter escrito a mensagem volte ao menu para escolher a nova opção.

Você pode apertar a tecla "4" se quiser "acabar" com o programa.

O programa pode guardar até 7 telas diferentes, porém apenas uma mensagem devido ao comprimento variável que a mesma pode ter.

Aí vai o programa

```

10 DIM t$(7,704)
20 LET c$ = "■"
30 CLS
40 POKE 23692,-1: PRINT AT 21,0;
50 PRINT INK 1; "1-Criar uma tela nova", "2-Apresentar a
tela existente", "3-Alterar a mensagem", "4-Pa-
rar"
60 INPUT k; CLS
70 IF k < 1 OR k > 4 THEN GOTO 60
80 IF k=4 THEN STOP
90 IF k=3 THEN GOTO 300
100 INPUT INK 2; FLASH 1; "Em que tela você quer desenhar?
(1-7)"; n
110 IF n < 1 OR n > 7 OR n<> INT n THEN GOTO 100
120 INPUT "Qual é a cor da tinta? □□□ "; t
130 IF k=2 THEN GOTO 390
140 PRINT INK 2; "Criar a tela usando as teclas 5,6,7 ou 8
para controlar o cursor"
150 PRINT ' INK 1; "Use 1 para ■, 2 para espaço em branco
e C para chega"
160 LET l = 11: LET c=16
170 LET t$(n,1 TO 704) = "□"
180 FOR z=1 TO 599: NEXT z
190 PRINT AT 0,0; INK t; t$(n); AT l,c; INK 1; FLASH 1; "+"
200 LET e$ = INKEY$: IF INKEY$ = "" THEN GOTO 200
210 IF e$ = "1" THEN LET c$ = "■"
220 IF e$ = "2" THEN LET c$ = "□"
230 LET c = c - (e$ = "5") + (e$ = "8")

```



```

240 LET c = c + (c<0) - (c>31)
250 LET l = l - (e$ = "7") + (e$ = "6")
260 LET l = l + (l<0) - (l>21)
270 LET t$ (n, l*32+c+1) = c$
280 IF e$ = "C" OR e$ = "c" THEN GOTO 300
290 GOTO 190

300 INPUT INK 1; "Em que linha você quer imprimir a mensa-
gem publicitária? (0-21)";ll
310 IF ll < 0 OR ll > 21 THEN GOTO 300
320 SOUND 1/(ll +1), ll: SOUND 1/(ll +10), ll+2
330 INPUT INK 1; "Qual é a cor?";cor
340 INPUT FLASH 1; "Você quer ver cintilando? (1 para sim e 0
para não)";f
350 SOUND 1/(f+2), f+4: SOUND 1/(f+4), f+8
360 INPUT INK 2; FLASH 1; "Entre com a sua mensagem..."; f$
370 LET l$ = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" + f$ +
"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
380 GOTO 30
390 PRINT AT 0,0; INK t; t$(n)
400 FOR j=1 TO LEN l$ - 32
410 SOUND .018,12
420 PRINT AT ll 0; INK cor; FLASH f; l$ (j TO j+31)
430 IF INKEY$ = "c" OR INKEY$ = "C" THEN STOP
440 PAUSE 40
450 NEXT j
460 GOTO 30

```

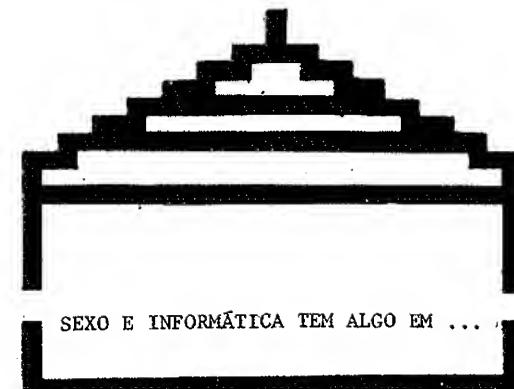


Figura 11



**TK 90X, o micro  
cheio de programas.**  
**MICRODIGITAL**

Execute o programa e veja se obtém algo do  
seguinte tipo:

# Capítulo 8

## Balança

Este é um programa útil principalmente para aqueles ou melhor para aquelas que tem o hábito de efetuar pesagens do tipo quanto vai isto paradar certo em uma receita (aindo acho que a maior parte do pessoal da cozinha é constituído por pessoas de sexo feminino...).

Na realidade o programa que vem a seguir é um tipo de jogo no qual com o menor número de tentativas e com a ajuda de um conjunto de "pesos" você deve dizer quantos necessita para equilibrar uma balança.

Ao executar o jogo em uma das suas etapas pode-se ter a situação indicada na Figura 12.

Na tela aparece uma balança com um peso colocado no prato da esquerda.

O TK 90X lhe pedirá que escolhas alguns entre os 6 pesos que aparecem na parte inferior da tela.

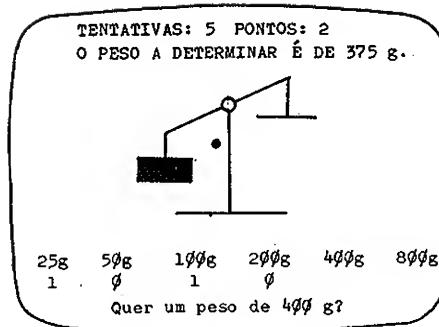


Figura 12

O TK 90X irá passando por todos os pesos disponíveis possibilitando-lhe escolher aqueles que desejas usar.

Quando quiseres usar um peso tecle "S" e em caso contrário um "N" (ou qualquer outro caractere).

Os seus pontos ficam registrados na parte de cima da tela.

10 REM \*\* O jogo da balança \*\*\*

20 RAND

30 LET c = Ø → acumula-se a contagem

40 LET t = Ø → contam-se as tentativas

50 CLS

60 LET k = 25

70 PRINT PAPER 6; INK Ø; AT Ø,2; "TENTATIVAS:"; t

80 PRINT PAPER 6; INK Ø; AT Ø,18; "PONTOS:"; c

90 FOR j=1 TO 30 STEP 5

100 PRINT PAPER 2; INK Ø; AT 15, j; k; "g"

110 LET k=k\*2



```

120 NEXT j
130 GOSUB 500
140 LET tinta = INT (RND * 5)+2
150 LET tt = 0
160 LET r = (INT(RND*63)+1)*25
170 PRINT PAPER 4; INK 0; AT 2,0; "O PESO A DETERMINAR E
    DE  "; r; "g"
180 LET t = t+1
190 PRINT AT 8,10; "█████"
200 LET d = 0: GOSUB 700
210 PRINT INK tinta; AT 10,10; "███"
220 LET k=25
230 FOR j=1 TO 6
240 PRINT PAPER 4; INK 0; AT 20,2; "Você quer um peso de  ";
    k; "g?"
250 LET r$ = INKEY$
260 IF r$ = "" THEN GOTO 250
270 IF r$ = "S" OR r$ = "s" THEN LET tt = tt+k: PRINT PAPER
    3; INK 0; AT 17, (j-1)*5+2; "1"
280 IF r$<>"S" AND r$ <>"s" THEN PRINT PAPER 6;
    INK 0; AT 17,(j-1)*5+2; "0"
290 FOR z=1 TO 45: NEXT Z: LET k=k*2: SOUND .018,10
310 NEXT j
320 PRINT INK 2; AT 3,18; "███ "
330 FOR z=1 TO 45: NEXT z
340 IF tt<r THEN GOTO 470
350 LET d=1: GOSUB 700
360 PRINT AT 10,10; "███"
370 PRINT AT 8,18; "███"
380 IF tt>r THEN GOTO 445
390 LET d = 0: GOSUB 550
400 PRINT INK tinta; AT 9,10; "███"
410 PRINT INK 2; AT 9,18."███"
420 SOUND .5,5: SOUND .6,6

```



```

430 LET c=c+1
440 GOTO 480
445 LET d=0: GOSUB 800
450 PRINT INK tinta; AT 8,10; "███"
460 PRINT INK 2; AT 10,18; "███"
470 SOUND 1,-13
480 FOR p=1 TO 300: NEXT p
490 GOTO 50
500 PLOT 95,80
510 DRAW 60,0
520 PLOT 125,80
530 DRAW 0,50
540 CIRCLE 125,130,4
545 RETURN
550 PLOT OVER d; 90,130
560 DRAW OVER d; 70,0
570 DRAW OVER d; 0,-30
580 PLOT OVER d; 150,100
590 DRAW OVER d; 20,0
600 PLOT OVER d; 90,130
610 DRAW OVER d; 0,-30
620 RETURN
700 PLOT OVER d; 90,120
710 DRAW OVER d; 70,20
720 DRAW OVER d; 0,-30
730 PLOT OVER d; 150,110
740 DRAW OVER d; 20,0
750 PLOT OVER d; 90,90
760 DRAW OVER d; 0,30
770 RETURN
800 PLOT OVER d; 90,140
810 DRAW OVER d; 70,-20
820 DRAW OVER d; 0,-30

```

```

83@ PLOT OVER d; 15@,9@
84@ DRAW OVER d; 2@,@
85@ PLOT OVER d; 9@,11@
86@ DRAW OVER d; @,3@
87@ RETURN

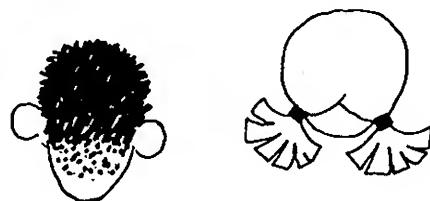
```



**O.I.-** Por favor após executar o programa se achar que o mesmo é infantil, não lhe é permitido falar impropérios visto que somar não é todo mundo que consegue hoje em dia e de cabeça ou melhor com a cabeça apenas...

Não esqueça que este livro nem passou pela censura pois ele só é impróprio para pessoas impróprias.

Mas você é o(a) próprio(a)!!!!



# Capítulo 9

## Química

Um (uma) "cientista louco" ("pirada") descobriu um líquido altamente perigoso o qual se não for misturado em proporção correta com a água fará explodir o seu TK 9@X, o que estiver nas suas proximidades e como não podia deixar de ser VOCEEEEEEE...

Você tem entretanto a possibilidade (e suponho a capacidade...) de impedir que isto aconteça colocando a quantidade de água correta que torne sem efeito o poder destruidor do fluido.

Para isto basta que você calcule (mentalmente...) quanta água é necessária para "desativar" o fluido empregando a relação que aparece na tela onde o primeiro número corresponde no fluido e o segundo a água.

Caso a sua resposta seja correta com aproximação de até 5% do verdadeiro valor você está salvo nesta porém se estiver fora deste intervalo ouvirás uma explosão..

Dizem que gato(a) tem "sete vidas" e neste jogo explosivo lhe são conferidas quatro vidas, após o que o jogo químico termina.

Entre pois com a quantidade de água que pensas ser a conveniente, tecle "ENTER" e note como vai caindo água dentro do recipiente e você nem abriu a torneira...

Bem, tenha vida longa, viu!!!

Aí vai o programa:

```

10 REM ** Químico **
20 RAND
30 LET C = 0
40 LET v = 4
50 LET f = INT(RND*5)+1
60 LET a = INT(RND*7)+2
70 PRINT PAPER 2; INK 0; AT 6,3; "Prepare-se, pois você tem:"
80 PRINT PAPER 3; INK 0; AT 8,1; f; "◻ partes de fluido por
◻"; a; "◻ partes de água".
90 PAUSE 300
100 CLS
110 PRINT PAPER 6; INK 0; AT 0,4; "vidas :◻"; v;
120 PRINT PAPER 6; INK 0; AT 0,19; "pontos:◻"; c
130 PRINT PAPER 2; INK 0; AT 3,10; "A RELAÇÃO É DE:◻"; f;
":"; a
140 PLOT 110,90
150 DRAW 0,-60
160 DRAW 35,0
170 DRAW 0,60
180 LET r = INT(RND*25)+1
190 PRINT PAPER 4; INK 0; AT 6,6; "Tens ◻ "; r; " ◻ litros
de fluido"
200 PRINT PAPER 4; INK 0; AT 8,2; "Quanta água você vai
usar?"
```

```

210 PRINT INK 2; AT 16,14; "■■■■"
220 PRINT INK 2; AT 17,14; "■■■■"
230 INPUT resposta
240 PRINT PAPER 4; INK 0; AT 20,8; resposta; "◻ litros de
água"
250 LET t = (r/f)*a
260 FOR j=1 TO 4
270 FOR k=10 TO 15
280 PRINT INK 5; AT k,16; "■"
290 PAUSE 7
300 PRINT AT k,16; "◻"
310 SOUND .15,k
320 NEXT k
330 FOR j=15 TO 12 STEP -1
340 PRINT INK 5; AT j,14; "■■■■"
350 NEXT j
360 PAUSE 150
370 IF resposta > t - t/20 AND resposta < t + t/20
    THEN GOTO 490
380 FOR j=6 TO 0 STEP -1
390 PAPER j
400 SOUND .25, j*2: SOUND .3,j*3
410 CLS
420 NEXT j
430 PRINT INK 7; AT 6,0; "Você tem agora uma vida a
menos!"
440 PAUSE 300
450 LET v=v-1
460 PAPER 7
470 IF v=0 THEN GOTO 570
480 GOTO 100
490 SOUND .3,7: SOUND .5,10: SOUND .7,12
500 FOR j=12 TO 17
510 PRINT INK 3; AT k,14; "■■■■"
```



**CAPS SHIFT** + tecla  
"9", quatro vezes a  
tecla "8" e novamen-  
te a tecla "9"

```

520 NEXT j
530 PRINT PAPER 4; INK 0; AT 20,3; "Fluido díluido correta-
mente"
540 LET c = c+1
550 PAUSE 350
560 CLS: GOTO 50
570 CLS
580 PRINT PAPER 3; INK 0; AT 9,3; "Perdeste todas as suas
vidas"
590 PRINT PAPER 3; INK 0; AT 12,6; "A sua pontuação final
é □"; c
600 STOP

```



No decorrer do jogo uma possível tela pode ser

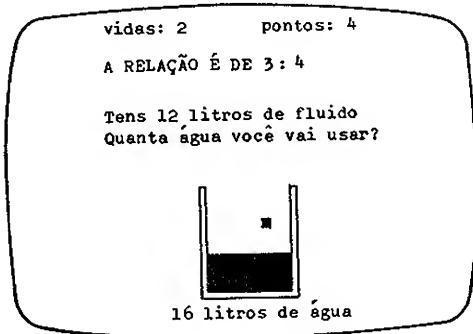


Figura 13

# Capítulo 10

## Probabilidades

Neste capítulo temos um programa que imprime uma tabela e também o gráfico mostrando os possíveis resultados do lançamento simultâneo de três dados quanto a soma de pontos obtidos.

Aliás, uma das coisas mais "elementares" ao se iniciar um curso de teoria das probabilidades é este tipo de discussão, visto que este modelo matemático permite definir com simplicidade o que vem a ser eventos mutuamente exclusivos, independentes, dependentes, etc.

O programa indica todos os casos quanto a soma de pontos, enumera os resultados favoráveis e imprime a probabilidade de cada um.

Além disto é apresentado na tela o gráfico da curva da distribuição de probabilidades (veja a Figura 14).

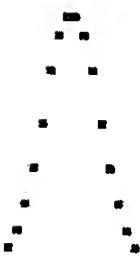


Figura 14

Aí vai o pequeno, porém importante programa.

```

10 LET j=RND*4+3
20 FOR s=3 TO 18
30 PRINT AT s,20; PAPER j; "■■■■■■■■■■■■■■■■"
40 NEXT s
50 DIM s(18)
60 PRINT AT 1,0; "SOMA"; TAB 5; "CASOS"; TAB 12; "PROB.";
    TAB 21; "GRÁFICO"
70 FOR b=1 TO 6
80 FOR c=1 TO 6
90 FOR d=1 TO 6
100 LET e=b+c+d
110 LET s(e) = s(e)+1
120 NEXT d: NEXT c: NEXT b
130 FOR s=3 TO 18
140 PRINT AT s,0; s; TAB 5; s(s); TAB 12; "1/"; INT
    (216/s(s))
150 PLOT INK 0; 2*s+185, 50+3*s(s)
160 NEXT s

```

Depois de um "programinha" deste (será que existe gente que é fanático(a) por matemática, mais especificamente pela teoria das probabilidades?) só um pouco de música clássica para acalmar aquele(a) que se transtornou...

Apesar de que a "onda" atual é enaltecer a Mozart (vide o sucesso de "Amadeus") o seu TK 90X irá homenagear através do programa aqui apresentado à Ludwig Von Beethoven, compositor alemão nascido em Bonn em 1770 e falecido em Viena em 1827.

Beethoven foi autor de trinta sonatas para piano, dezessete quartetos, nove sinfonias, cinco concertos para piano e um para violino.

Além disto é também o autor de aberturas da ópera Fidélio, da Missa solemnis (missa em ré), obras de profundezas de sentimentos e de um poder de expressão artística incomparáveis.

Beethoven teve existência muitas vezes difícil e foi, cedo, atacado de surdez.

Este programa é recomendável a todos que gostam de música clássica e principalmente para aqueles que sejam seguidores do "ilustre surdo".

Além do que, não há quem não goste de ser despertado(a) ou estar envolvido(a) por uma melodia como a que se vai ouvir!!!

```

10 DIM s(122)
20 FOR j=1 TO 122
30 READ s(j)
40 NEXT j
50 FOR j=1 TO 122
60 SOUND .22,s(j)
70 NEXT j
80 GOTO 50

```



```

100 DATA 16,15,16,15,16,11,14,12,09,04,09,00,04,09,11,
    04,08,04,08
110 DATA 11,12,04,09,04
120 DATA 16,15,16,15,16,11,14,12,09,04,09,00,04,09,11,
    04,08,04,12
130 DATA 11,09,04,09,11,12,14,16,07,12,07,17,16,14,07,
    11,07,16,14
140 DATA 12,4,9,4,14,12,11,4,8,4,8,11,16,16,15,16,15,16,
    11,14,12,9,4,9,0,4,9,11,4,8,4,8,11,12,4,9,4
150 DATA 16,15,16,15,16,11,14,12,9,4,9,0,4,9,11,4,8,4,
    12,11,9,4,9

```



# Capítulo 11

## Fahrenheit x Celsius

Neste programa exibe-se a conversão de graus centígrados ou Celsius para graus Fahrenheit e vice-versa tanto analiticamente como graficamente cabendo a você uma das escolhas ou ambas.

A cada resultado numérico o seu TK 90X adiciona o seu sistema de "ruído" para não dizer os recursos sonoros.

Note o recurso que lhe permite o TK 90X de criar o caractere "°" para indicar os graus (linha 160).

```

10 REM *** Fahrenheit X Celsius ***
20 PAPER 2: INK 5: BORDER 1
30 FOR j=0 TO 7: READ código
40 POKE USR "a" + j, código: NEXT j
50 DATA 64,160,64,0,0,0,0,0
60 REM * Para obter o caractere da linha 160 escreva "a" em
    modo gráfico *

```

```

70 PRINT "Faça a sua escolha"
80 PRINT: PRINT
90 PRINT "Entre com 1 para a conversão centígrados para
      Fahrenheit"
100 PRINT
110 PRINT "Entre com 2 para a conversão Fahrenheit para
      centígrados"
120 INPUT n
130 IF n=1 THEN GOTO 180
135 IF n<>2 THEN GOTO 120
140 INPUT "Entre com os graus Fahrenheit"; f
150 LET c=5*(f-32)/9
160 PRINT f; " °Fahrenheit = "; c; " °centígrados"
170 GOTO 210
180 INPUT "Entre com os graus centígrados"; c
190 LET f=1.8*c+32
200 PRINT c; " °centígrados = "; f; " °Fahrenheit"
210 PRINT: PRINT
220 PRINT "Você quer fazer alguma outra conversão? (S/N)"
230 INPUT r$
240 IF r$ = "s" OR r$ = "S" THEN GOTO 80
250 SOUND 2,20: SOUND 1.5,15: SOUND 1,10: SOUND .5,5
260 CLS
270 PRINT FLASH 1; "Você agora vai apreciar uma interessante
      conversão gráfica com a escala Fahrenheit na vertical e
      a escala Celsius na horizontal"
280 REM ** Aí vai o gráfico de conversão que evidentemente é
      aproximado
290 FOR f = 30 TO 220 STEP 10
300 IF f < 100 THEN PRINT "□";
310 PRINT f;
320 FOR c=1 TO 5*(f-32)/45
330 IF c/6-1=INT(c/6-1) THEN LET t=0
340 LET t = t+1

```



```

350 PRINT INK t; "□";
360 NEXT c
370 LET t=0: PRINT
380 NEXT f
390 PRINT TAB 3; "102030405060708090100"
400 FOR j=10 TO 20: SOUND .1,j
410 SOUND 1,20-j: NEXT j
420 STOP

```

Ao executar a 2ª parte do programa (linhas 280 a 410) você deve obter o seguinte desenho:

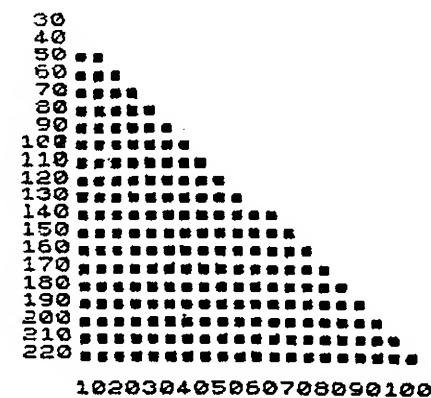


Figura 15

# Capítulo 12

## Aeroplano

Vamos executar um programa que permita movimentar o aeroplano da Figura 16 da esquerda para a direita da tela.

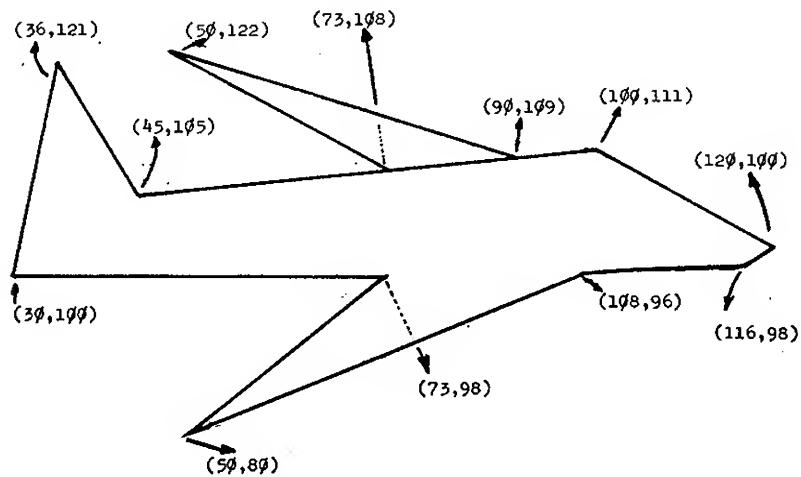


Figura 16

Vai se desenhar a próxima posição antes de apagar a imediatamente anterior.

Ao teclar e executar o programa que vem a seguir você vai se convencer de que a animação conseguida com programação em linguagem BASIC onde se utiliza fundamentalmente a instrução DRAW é bastante lenta para que se possa dizer que temos realmente um movimento.

Para que se possa "animar" figuras do tamanho da indicada na Figura 16 a solução adequada é o uso de linguagem de máquina.

Aliás, o meu filho Victor promete para breve um livro sobre o assunto...

5 REM \*\*\* Movimento do aeroplano \*\*\*

10 PAPER 7: INK 1:: BORDER 7

20 FOR j=120 TO 240 STEP 2

30 PLOT j, 100: GOSUB 100

43 CLS

45 PLOT j,100: GOSUB 100

47 SOUND 0.2,-j/60

50 NEXT j

53 INK 6\*RND

55 FOR Z=1 TO 399: NEXT Z

60 CLS

70 GOTO 20

100 DRAW -20,11

110 DRAW -10,-2

120 DRAW -40,13

130 DRAW 23,-14

140 DRAW 17,1

150 PLOT j-47,108

160 DRAW -28,-3

170 DRAW -9,16



```

180 DRAW -6,-21
190 DRAW 43,-2
200 DRAW -23,-18
210 DRAW 35,13
220 DRAW 25,1
230 DRAW 10,6
240 RETURN

```

Se você gostou deste programa escolha um sistema de referência adequado e faça se mover de cima para baixo e vice-versa o seguinte helicóptero (Figura 17).

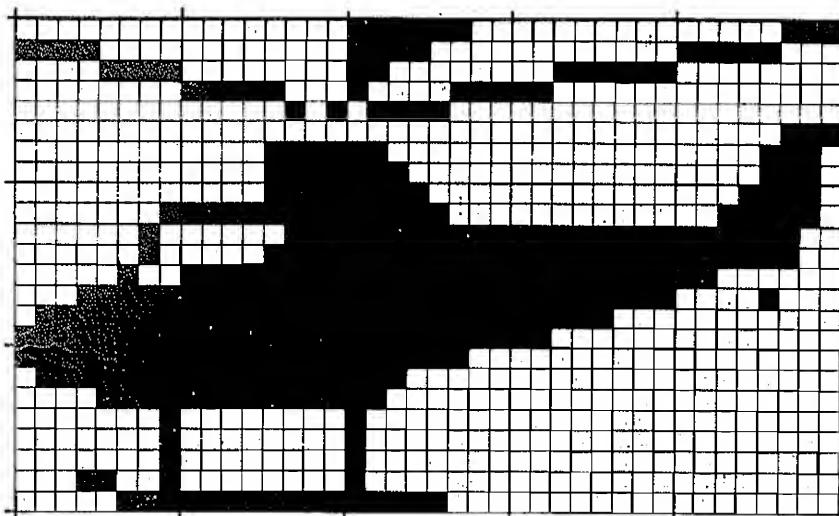


Figura 17

É natural que não espero que o seu helicóptero seja tão "lindo" como o da Figura.

Mãos a obra, veja se consegue isto!!!!.

## Capítulo 13

### Lote econômico

Este é um programa de aplicação comercial no qual se pode calcular o lote econômico a encenhar para se efetuar a reposição de estoque ou seja é a quantidade de unidades de um certo artigo que tornam mínimo o custo global levando-se em conta apenas o custo de manutenção do estoque e o custo do pedido que implica entre outras coisas em pagar o transporte.

O seu TK 90X lhe pedirá a demanda para um período de planejamento, o número que corresponde ao código de um artigo o preço unitário do mesmo, o nome do seu fornecedor e o custo de um pedido.

**O.I.** - Se você tirar muitas dúvidas sobre o assunto consulte o nosso livro Pesquisa Operacional - vol.1 ou então Aplicações de Pesquisa Operacional - vol.1.

Aí vai o programa:

```

5 PAPER 7: INK 1: BORDER 7
10 REM d = demanda para o horizonte de planejamento
20 REM r = código do artigo
30 REM C = preço unitário do artigo
40 REM f$= nome do fornecedor
50 REM k = custo de um pedido
60 REM h = custo de estocagem de um artigo por período
70 REM h' = .20*C (fórmula adotada)
80 INPUT "Entre com a sua demanda";d
90 INPUT "Entre com o código do artigo";r
100 INPUT "Entre com o preço unitário do artigo";C
110 INPUT "Entre com o nome do fornecedor";f$
120 INPUT "Entre com o custo do pedido"; k
130 REM ** Cálculo do lote econômico**
140 LET h' = .2*C
150 LET y = SQR(2*k*d/h): LET y = INT(y+.5)
160 PRINT: PRINT
170 PRINT TAB 5; "Você deve pedir de";f$
180 PRINT: PRINT
190 FOR z=1 TO 32
200 PRINT "*";
210 NEXT z
220 PRINT TAB 3; y;" unidades do artigo#";r
230 FOR z=1 TO 32
240 PRINT "*";
250 NEXT z
260 LET custo=SQR(2*k*h*D): LET custo=INT(100*custo+.5)/100
270 PRINT "O custo destes pedidos para o horizonte de demanda
d ="; d; " artigos é:"
280 PRINT
290 PRINT TAB 10; "custo=";custo
300 PRINT

```



```

310 INPUT "Você quer ver a representação gráfica do seu
problema de estoque (S/N)?"; r$
320 IF r$ = "s" OR r$ = "S" THEN CLS: GOTO 340
330 STOP
340 REM ** Aí vai a construção das curvas de custo
345 LET a=1: LET b=y+30
350 FOR j=a TO b
360 PLOT 5*j,k*d/(60*j)
370 PLOT !5*j,h*j/120
380 PLOT 5*j, 1/60*(k*d/j+h*j/2)
390 NEXT j
400 PLOT 0,175: DRAW 0,-175: DRAW 255,0
410 FOR k=1 TO 10
420 SOUND .2,11
430 NEXT k
440 GOTO 330

```

este 60 é um fator de escala que o cara(a) leitor (leitora) pode mudar caso seja necessário

Execute e veja se obtém um gráfico do seguinte tipo:

Aplique para:

$d = 300$   
 $C = 200$   
 $k = 30$

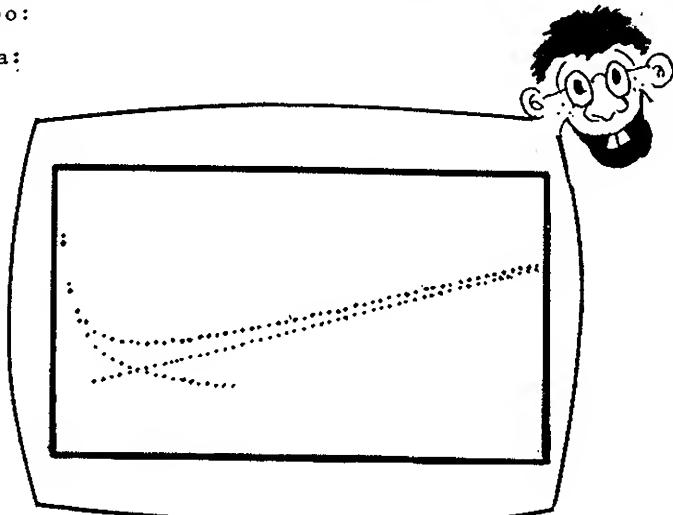


Figura 18

# Capítulo 14

## Chapéu mexicano

Quem é que não fica fascinado quando um micro lhe consegue uma representação a três dimensões?

Desconheço...

É muito simples, em certos casos, obter representações tridimensionais quando se tem as funções matemáticas.

É o caso deste programa no qual se representa a função

$$z = e^{-r/3} \cdot \cos r$$

onde r é a distância do centro do plano X-Y até o ponto que se está desenhando.

Embora o programa seja relativamente pequeno para entrar no TK 90X esteja ciente que vai levar quase meia hora para que você veja o que está mostrado na Figura 19.

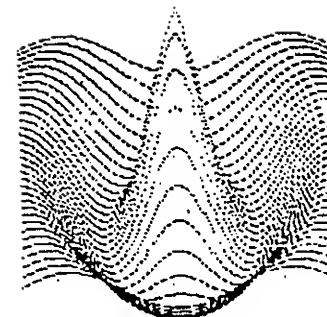


Figura 19

```

10 FOR x=40 TO 215
20 LET max = 999: LET min = 0
30 FOR y = 16 TO 144 STEP 4
40 LET r=SQR((x-127)*(x-127)+(y-80)*(y-80))/15
50 LET z=INT(y+90*EXP(-r/3)*cosr)
60 IF z < max OR z > min THEN PLOT x,z
70 IF z < max THEN LET max = z
80 IF z > min THEN LET min = z
90 NEXT y
100 NEXT x

```



Aí vêm algumas explicações:

Linhas 10 a 100 → É o "LOOP" para todos os pontos com x variando de 40 até 215.

Linha 20 As variáveis "max" e "min" são utilizadas para não permitir representações fora dos limites.

Linhas 30 a 90 → LOOP com passo 4 para a variável y de 16 até 144.

Linhas 40 e 50 → Calcula-se a altura na qual deve-se representar o ponto.

Linha 60 → "Desenha-se" desde que não se esteja fora dos limites.

Linhas 70 e 80 → Atualizam-se os valores das "variáveis" max e min ou seja dos pontos mais alto e mais baixo para esta coordenada x.

Bem, execute agora o programa, mas tenha bastante paciência...

Ninguém é de ferro, não é?

a Eu sei disso!!!

É justamente por isto você agora vai ter um pouco de diversão associada a uma interessante representação gráfica na tela.

Nada mais merecido para quem espera tanto para ver um "sombrerão". Seja porém bem dinâmico no jogo "Brincando com a cobra".

O intuito do jogo é fazer com que uma cobra sinuosa e sua amiga (por incrível que possa parecer...) consiga passar entre um conjunto de triângulos.

Caso a cobra toque em algum destes triângulos com a sua cabeça receberá um ferimento mortal e com isto você perde a sua amiga e o jogo termina...

Como fazer para não matar a sua amiga?

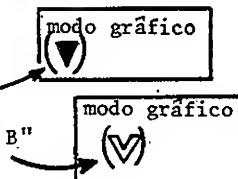
Bem, você perceberá logo que a cobra se desloca para a esquerda sem a necessidade de nenhum comando (até parece uma cobra política...) porém você pode fazer com que ela vá para a direita pressionando a tecla "8".

Bem divirta-se agora e espero que consiga marcar mais que 1000 pontos...

```

10 REM *** A cobra nos triângulos ***
20 GOSUB 310: LET recorde = 0
30 GOSUB 270
40 PRINT AT 9, c; "□"
50 LET c = c+2*(INKEY$ = "8" AND v < 29)
60 IF c > 0 THEN LET c = c-1
70 IF SCREEN$(10,c) = "" THEN GOTO 200
80 PRINT AT 9,c; INK 6; "A"; AT 10, c; "B"
90 PRINT AT 21,0;
100 PRINT TAB INT(RND*31); INK 1; "C"
110 POKE 23692,255
120 LET p=p+1: PRINT
130 GOTO 40
200 PRINT AT 0,0; OVER 1; PAPER 8; INK 2; a$(1)
210 PRINT AT 1,10; FLASH 1; "O jogo acabou"
220 PRINT AT 6,5; "Você obteve:"; p; "□ pontos"
230 IF p > recorde THEN LET recorde = p
240 PRINT AT 8,3; "Seu recorde até agora é:□"; recorde;
      "□ pontos"
250 INPUT "Aperte □"; PAPER 1; "ENTER"; PAPER 0;"□ para
      jogar de novo □"; LINE r$: GOTO 30
260 STOP
270 BORDER 0: PAPER 0: INK 7: CLS: LET v=10: LET c=15
280 LET p=0: RAND:
290 DIM a$(1,704)
300 RETURN
310 FOR j= USR "a" TO USR "c" + 7
320 READ usuário: POKE j, usuário
330 NEXT j
340 RETURN

```


```

350 DATA 255, 126, 126, 60, 60, 24, 20, 0
360 DATA 0, 99, 119, 127, 62, 28, 0, 0
370 DATA 99, 119, 73, 73, 127, 62, 28, 0

```



# Capítulo 15

## Bioritmo

Este programa ilustra muito o uso de subrotinas quando uma operação similar ou geralmente a mesma é necessária mais do que uma vez em um programa.

Este programa se baseia na teoria de que os seus ciclos *emotivo, físico e intelectual* seguem um padrão regular de períodos 28, 23 e 33 dias respectivamente a partir da data do seu nascimento.

O programa "BIORITMO" permite que os "altos e baixos" dessas três curvas (veja a Figura 20) sejam representados para um período de 4 semanas centrado em qualquer data que você especificar.

Diz-se que os dias nos quais uma curva está cruzando a linha central são críticos.

Pode até "pintar" para você, que através de um estudo cuidadoso deste tipo de gráfico algo seja revelado sobre o seu comportamento ou talvez mesmo uma forma razoável de prognosticar o seu futu

mesmo uma forma razoável de prognosticar o seu futuro...

Como alerta para seus futuros programas coloridos e para lhe poupar experimentação inútil anote que comumente é bom ter um fundo preto e na maior parte dos casos as cores magenta (lilás), verde, ciano e amarela dão desenhos melhores que com as cores azul ou vermelha.

Também linhas diagonais muito finas podem provocar efeitos muito esquisitos e neste programa as curvas foram desenhadas com 3 pixels de largura para evitar este problema!!!

Quando você estiver executando este programa perceberá que toda vez que uma linha curva estiver sendo desenhada perto de uma outra linha curva desenhada anteriormente de cor diferente então parte da linha curva antiga mudará para a mesma cor que tem a linha curva nova.

Isto realmente é uma pequena limitação do TK 90X visto que ele só pode ter duas cores papel (PAPER) e tinta (INK) em qualquer casela para caractere.

Este programa se utiliza de duas subrotinas uma começando na linha 400 e terminando na linha 490 e a outra começando na linha 500 e terminando na linha 530.

A primeira (de 400 a 490) faz a análise da data em que você nasceu e a outra desenha as curvas senoidais que correspondem ao seu bioritmo.

Aí vai o programa:

```

10 PAPER 7: INK 0: BORDER 7: INVERSE 0: OVER 0: FLASH 0: CLS
20 PRINT AT 5,11; INK 3; "BIORITMO"
25 INPUT "Qual é o seu nome?"; LINE n$
30 PRINT AT 8,0; "Como vai"; n$; AT 10,0; "Qual é a data
do seu nascimento?"
40 GOSUB 400
50 PRINT AT 11,19; d$: LET b$ = d$; LET z=x
60 PRINT AT 12,0; "Qual é a data de interesse?":GOSUB 400
70 PRINT AT 13,17; "□□"; d$: LET dia = x-z
80 INK 3: PRINT AT 15,0; "Nesta data"
90 PRINT "Você estará □"; dia; "□ dias mais velho,"
100 PRINT "terá comido □"; 3*dia; "□ vezes"
110 PRINT "e terá dormido (aproximadamente)"
120 PRINT FLASH 1;TAB 15;INT(8*dia/1000); "□ milhares de horas"
130 INPUT "Aperte a tecla ENTER se você quer o seu gráfico □";
LINE e$
140 PAPER 0: INK 6: BORDER 0: CLS
150 PRINT TAB 5; "GRÁFICO DO SEU BIORITMO"
160 PRINT TAB 15 - LEN n$/2; n$
170 PRINT TAB 12 - LEN b$/2; "nascido em □"; b$
180 FOR i=1 TO 255 STEP 9: PLOT i,73: DRAW 0,3: NEXT i
190 FOR j=1 TO 255 STEP 63: PLOT j,71: DRAW 0,6: NEXT j
200 PLOT 127,10: DRAW 0,128
210 PRINT AT 21,0; "-2 semanas"; TAB 22; "+2 semanas"
220 PRINT AT 21,16 - LEN d$/2; d$
230 INK 3: PRINT AT 3,0; "EMOTIVO": LET c=28: GOSUB 500
240 INK 4: PRINT AT 3,10; "FÍSICO": LET c=23: GOSUB 500
250 INK 5: PRINT AT 3,21; "INTELECTUAL": LET c=33: GOSUB 500
260 PAPER 7: INK 0: BORDER 7
270 FOR j=1 TO 21 STEP 3
280 SOUND j/20,j
290 NEXT j
300 FOR j=21 TO 1 STEP -3
310 SOUND j/20,-j

```



```

320 NEXT j
330 STOP
400 DATA 0,31,28,31,30,31,30,31,31,30,31,30,31
410 INPUT "Qual é o ano?"; ano
415 LET x = 365*ano + INT(ano/4) - INT(ano/100)
420 INPUT "Qual é o mês (1-12)?"; mes
425 IF mes < 1 OR mes > 12 THEN GOTO 420
430 RESTORE 400
440 FOR j=1 TO mes
445 READ v: LET x=x+v: NEXT j
450 LET l = ano - 4*INT(ano/4)
455 IF l = 0 AND mes > 2 THEN LET x=x+1
460 READ v
465 IF l = 0 AND mes = 2 THEN LET v = 29
470 INPUT "Em que dia (1-31)?"; dia
475 IF dia < 1 OR dia > v THEN GOTO 470
480 LET x=x+d.
485 LET d$ = STR$ dia + "/" + STR$ mes + "/" + STR$ ano
490 RETURN
500 FOR k=0 TO 253: LET p = dia - 14+k/9
510 PLOT k, 74+60*SIN(2*PI*p/c): DRAW 2,0
520 NEXT k
530 RETURN

```



#### Algumas explicações sobre o programa

- 1) Linhas 10 até 70 → Pede-se aí a data do seu nascimento e a data atual, usando a subrotina que começa na linha 400.

Calcula-se o número de dias (d) entre as duas datas e armazena-se nas STRINGs b\$ e d\$ as duas datas.

É interessante destacar o efeito da instrução LINE n\$ (linha 25).



A instrução LINE é usada nos comandos INPUT ou SAVE.

Ao se usar a instrução LINE são omitidas as aspas durante a introdução da STRING porém consideram-se as eventuais aspas do texto.

Quando se usa LINE em conjunto com SAVE o programa é executado automaticamente a partir da linha determinada após à LINE.

- 2) Linhas 80 até 120 → São impressos aí algumas informações que talvez você identifique como interessantes...
- 3) Linha 130 → Aí o TK 90X espera que você lhe dê uma ordem para mostrar-lhe o gráfico do seu "bioritmo".
- 4) Linhas 140 até 220 → Com este trecho são impressos o cabeçalho, um sistema de referência.
- 5) Linhas 230 até 250 → Desenham-se devido a este trecho as três curvas usando para isto a subrotina (500-530).
- 6) Linhas 260 até 330 → Restaura-se as cores do papel (PAPER), da tinta (INK) e da borda (BORDER) como cortesia para o(a) próximo(a) usuário(a). Porém você deve aguentar um pouco de ruído

- 7) Linha 485 → Aí aparece uma concatenação efetuada com o auxílio da instrução STR\$ que transforma um argumento numérico em uma STRING.

Execute o programa e veja se obtém algo do tipo mostrado na Figura 21.

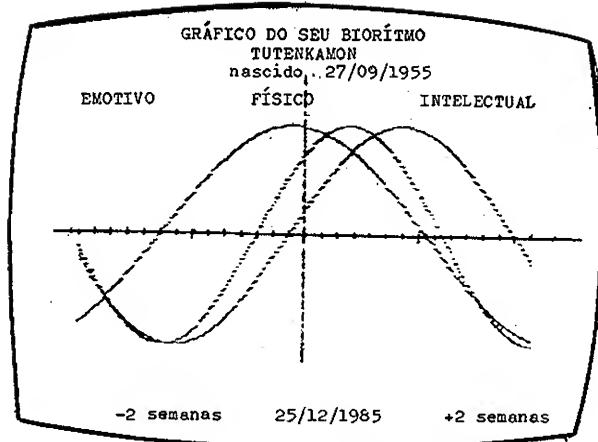
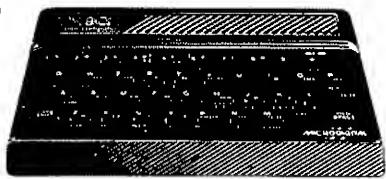


Figura 20



**TK 90X, o micro  
cheio de programas.**  
MICRODIGITAL



# Capítulo 16

## Representação de uma função

Este programa permite que o TK 90X faça o gráfico da função definida por você estimado(a) usuário(a).

Cabe a você escolher os limites entre os quais deseja desenhar a equação.

Uma vez executado, isto é após o RUN, o programa lhe perguntará a equação (como função de  $x$ ) que você quer que apareça na tela.

Não há condição neste programa de verificar se a função que você introduziu está correta por isto caro(a) leitor(leitora) tenha um cuidado extremo para não introduzir caracteres ilegais ou incorretos, falta de parênteses, etc.

O(A) usuário(a) deve teclar depois os limites mínimo e máximo do valor de  $x$  (abscissa) que deseja.

O programa também lhe perguntará pelo valor das assíntotas, caso existam.

Você sabe o que é uma assíntota?  
Não!?

Aí vai a resposta!

Uma assíntota é o valor de  $x$  para o qual a função toma um valor igual a  $\pm\infty$  (infinito).

É o caso típico da função  $y = \log x$  para  $x = 0$  quando  $y$  vale ou melhor tende a  $-\infty$ .

O mesmo ocorre com  $y = \frac{-1}{x-1}$  para  $x = 1$ . ou ainda com

$$y = \tan x \text{ para } x = \frac{\pi}{2}, \frac{3\pi}{2}, \dots, \frac{(2k+1)\pi}{2},$$

Se a função escolhida pelo(a) usuário(a) tem uma ou mais assíntotas, o programa perguntará pela tolerância no cálculo para desta forma se evitar o cálculo do valor da função para aqueles valores de  $x$  que estão próximos das assíntotas, o que inevitavelmente aconteceria se a tolerância fosse zero ( $0$ ).

Em outras palavras, se existe uma assíntota em  $x=2$  e a tolerância que você introduzir for  $0,01$  o programa não calculará os valores da função para  $x$  no intervalo de  $1,99$  a  $2,01$ .

O programa calcula a função para 211 valores de  $x$  dentro do intervalo definido, escolhendo os limites convenientes para o desenho dos eixos coordenados ( $x$  e  $y$ ).

As subrotinas que determinam os valores dentro de cada intervalo (linhas 580 a 710) utilizam um LOOP 5 vezes consecutivas, uma vez para cada intervalo.

O valor máximo calculado no LOOP é um pouco maior que o máximo requerido, devido a necessidade de que os erros acumulados devido ao arredonda-

mento não façam terminar o desenho do quarto intervalo dentro do espaço do quinto intervalo.

Analise a seguinte subrotina que ilustra o que ocorreria em caso contrário

```
10 INPUT N
20 FOR j=0 TO N STEP N/4
30 PRINT j
40 NEXT j
```



Se o valor de  $N$  com o qual você entrar for  $N = 2,01$  os resultados impressos no programa serão:

0; 0,5025; 1,005; 1,5075; 2,010.

Já se você entrar com o valor de  $N = 2,03$  então o programa escreverá:

0; 0,5075; 1,0150; 1,5225

Neste segundo caso só estão escritos 4 resultados porque os erros de arredondamento se acumulam no valor de STEP (incremento), agregando-se ao valor da variável  $j$  em cada etapa.

Depois de desenhar o gráfico, o programa pede ao(a) usuário(a) para introduzir as "coordenadas" para o comando PRINT AT, utilizado para escrever a expressão matemática cujo gráfico você está vendo na tela (linha 790).

Caso você não esteja satisfeito com a posição escolhida pode mudar a mesma simplesmente escrevendo por cima da própria através da utilização sucessiva de OVER 1 (linha 820).

Insistir em um único assunto durante muito tempo pode sugerir a você respeitado(a) leitor(leitora) falta de imaginação de nossa parte ou seja que estou o(a) considerando com poucos neurônios.

Vamos então direto ao programa que o resto das dúvidas serão tiradas por você mesmo meu(minha) caro(a)...

```

10 PRINT TAB 5; "DESENHO DE UMA FUNÇÃO"
20 PRINT: PRINT
30 PRINT "Entre com a função"
40 INPUT f$: PRINT,, "A função é:"; f$
50 PRINT,, "Entre com os valores mínimo e
máximo para poder representar a f(x)"
60 PRINT TAB 0; "xmin = ",: INPUT xi: PRINT xi
70 PRINT TAB 0; "xmax= ",: INPUT xf: PRINT xf
85 INPUT as: IF as < 0 THEN SOUND 1,13: GOTO 80
80 PRINT,, "Introduza o número de assíntotas"
90 IF as = 0 THEN GOTO 165
100 DIM a(as)
110 PRINT,, "Introduza os valores de x para cada assíntota
pela ordem"
120 FOR j=1 TO as
130 INPUT a(j)
140 NEXT j
150 PRINT,, "Introduza a tolerância permitida para a(s)
assíntota(s), por exemplo 0.01"
160 INPUT tol: IF tol < 0 THEN SOUND 1,11: SOUND 1,12: SOUND
1,13: GOTO 160
165 PAPER 5: INK 1: BORDER 1
170 PRINT AT 21,2; FLASH 1; "Estou fazendo os cálculos!?""
180 DIM x(211): DIM y(211)
190 LET i=1
200 FOR x=xi TO xf STEP (xf-xi)/210
210 IF as = 0 THEN GOTO 250
220 FOR j=i TO as
230 IF ABS(x-a(j)) < tol THEN GOTO 280
240 NEXT j
250 LET x(i) = x

```



```

260 LET y(i) = VAL f$
270 LET i=i+1
280 NEXT x
290 LET n=i-1
300 LET yi = y(1)
310 LET yf = y(1)
320 FOR i=1 TO n
330 IF yi > y(i) THEN LET yi = y(i)
340 IF yf < y(i) THEN LET yf = y(i)
350 NEXT i
360 LET yi = yi * .99
370 LET yf = yf * 1.01
380 CLS
390 PAPER 4: INK 1: BORDER 3
400 LET x=35: LET y=11
410 DRAW INVERSE 1; OVER 1; x,y
420 FOR k=1 TO 4
430 DRAW 0,41
440 DRAW -3,0
450 DRAW 3,0
460 NEXT k
470 DRAW 0,-164
480 DRAW -3,0
490 DRAW 3,0
500 DRAW 0,-3
510 DRAW 0,3
520 FOR k=1 TO 4
530 DRAW 55,0
540 DRAW 0,-3.
550 DRAW 0,3
560 NEXT k
570 DRAW -228,0
580 LET a=4

```



```

590 FOR z=xi TO xf STEP (xf-xi)/4
600 LET z$ = STR$ z
610 IF LEN z$ > 4 THEN LET z$ = z$(TO 4)
620 IF a = 31 THEN LET a = 32 - LEN z$
630 PRINT AT 21,a; z$
640 LET a = a+6 + (a > 9)
650 NEXT z
660 LET a = 20
670 FOR t=yi TO 1.01 * yf STEP (yf-yi)/4
680 LET z$ = STR$ t + "□"
690 PRINT AT .a,0; z$(TO 4)
700 LET a = a-5
710 NEXT t
720 FOR i = 1 TO n
730 LET x = 35 + 210 * (x(i)-xi)/(xf-xi)
740 LET y = 12 + 163 * (y(i) - yi)/(yf-yi)
750 PLOT x,y
760 NEXT i
770 INPUT "Dê a posição do título"; l;"□"; c
780 IF l > 0 OR l > 21 OR c < 0 OR c > 31 THEN SOUND 1,12;
    GOTO 720
785 PAPER 5
790 PRINT OVER 1; AT l ,c; f$
800 INPUT "Está correto? (S/N)"; r$
810 IF r$="S" THEN SOUND 1,22; STOP
820 IF r$ = "N" THEN PRINT OVER 1; AT l,c; f$: GOTO 770
830 IF r$<>"S" THEN GOTO 800

```



Veja se entrando com a função  $\frac{\sin(2x)}{x}$  sai algo do tipo do que está mostrado na Figura 21).

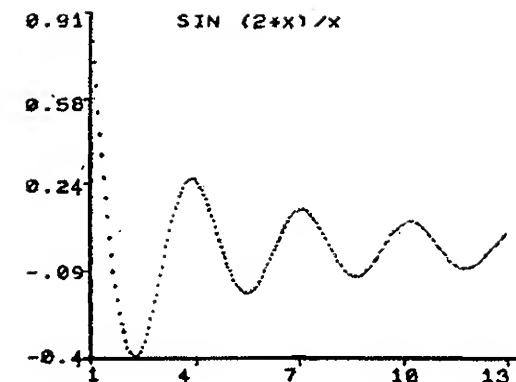


Figura 21



# Capítulo 17

## Espirais

"ESPIRAIS" é um programa para tornar exponencial a sua habilidade, o seu talento artístico e o seu gênio criativo meu(minha) caro(a) amigo(a) proprietário(a) deste livro e do TK 90X.

Não existe nenhum limite para a variedade de curvas e espirais que podem ser desenhadas com este programa, salvo as impostas pela imaginação do (a) usuário(a)...

O programa é tremendamente auto-instrutivo ou seja totalmente tutorial fornecendo-lhe todas as instruções necessárias para escolher e fazer com que o TK 90X desenhe cada uma das seis curvas exemplo.

No decorrer da execução, o programa cria uma cadeia literal (STRING) f\$, na qual as equações das curvas que se quer desenhar se armazenarão de forma permanente.

Em continuação são lidos os 6 pares de equações (as que vão como exemplo dentro das instruções DATA) enviando-os para um armazenamento temporário em x\$ e y\$.

O programa se transfere para as equações que estão em x\$ e y\$, vai a subrotina de número de linha 1500, desenha a curva escolhida e continua fazendo isto até que o(a) usuário(a) aperte alguma tecla quando então se volta ao "menu" de opções

MENU
Introduzir uma função → I
Desenhar uma função → D
Armazenar uma função → A
Parar o programa → P

Tabela 1

Aí vão alguns comentários sobre as curvas exemplo.

É surpreendentemente difícil prever o tamanho de uma curva a partir das suas equações paramétricas.

A circunferência é a curva fundamental ou seja serve de "origem" para todas as curvas fechadas, elipses e espirais.

As equações paramétricas de uma circunferência de raio unitário são:

$$\begin{cases} x = \cos t \\ y = \sin t \end{cases}$$

Por outro lado, caso se queira uma circunferência de raio, digamos 75 unidades, basta ter as equações

$$\begin{cases} x = 75 \cos t \\ y = 75 \sin t \end{cases}$$

A maior circunferência que você pode desenhar na tela do aparelho de T.V. conectado ao seu TK 90X é de raio 87.

Pode-se pensar em uma elipse como sendo "aproximadamente" uma circunferência de círculo que foi alongada ou achatada.

A maior elipse que pode ser desenhada com centro no centro da tela ( $x = 127$ ,  $y = 87$ ) tem como equações paramétricas:

$$\begin{cases} x = 127 \cos t \\ y = 87 \sin t \end{cases}$$

Para que se possa girar uma elipse de um certo ângulo é necessário "mesclar" as funções trigonométricas seno e cosseno nas equações paramétricas, como por exemplo nas equações:

$$\begin{cases} x = 4\theta \cos t + 25 \sin t \\ y = 4\theta \sin t - 25 \cos t \end{cases}$$

Efeitos extremamente interessantes são obtidos quando se "misturam" duas elipses (veja a Figura 22).

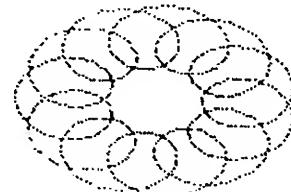


Figura 22

É o caso, quando tivermos as equações paramétricas

$$\begin{cases} x = 25 \cos(12t) + 5\theta \cos t \\ y = 15 \sin(12t) + 3\theta \sin t \end{cases}$$

Figura 22

Outros desenhos "fantásticos" podem ser obtidos com as equações paramétricas

$$\begin{cases} x = 7\theta \cos(t/8) \cos t \\ y = 5\theta \sin(t/8) \sin t \end{cases}$$

Figura 23

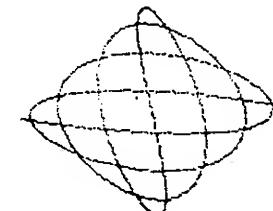


Figura 23

$$\begin{cases} x = 15 \cos(10t) + 8\theta \cos t + 5\cos(t/4) \\ y = 15 \sin(10t) + 45 \sin t + 5\sin(t/4) \end{cases}$$

Figura 24

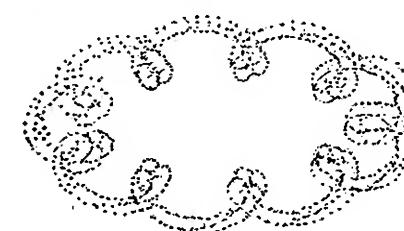


Figura 24

Bem aí vai o programa "ESPIRAIS"

```

5CLS: PRINT TAB 10; "ESPIRAIS"
10 GOSUB 2000
20 BORDER 6: PAPER 6: INK 1
30 LET fant=0: LET cf = 0: LET lf = 1000
40 DIM f$(lf)
50 DATA 6
60 DATA "15*COS(10*t) + 8\theta*COS t + 5*COS(t/4)", "15*SIN(10*t)
+ 45*SIN t + 5*SIN(t/4)"
70 DATA "7\theta*COS(t/8)*COS t", "5\theta*SIN(t/8) * SIN t"

```

```

80 DATA "25*COS(12*t) + 50 *COS t", "15*SIN(12*t)+30*SINT"
90 DATA "40*COS t +25*SINT", "40*SIN t - 25*COS t"
100 DATA "20*COS t + 20 * COS(t/2) + 20 * COS (t/4)",
    "15*SIN t + 15 * SIN(t/2) + 15*SIN (t/4)"
110 DATA "40*SIN t + 20*SIN(4*t) + 10*SIN (16*t) + 5*SIN(32*t)",
    "32*COS t + 8*COS(4*t)+ 4*COS(16*t) + 2*COS(32*t)"
120 READ nf
130 FOR j=1 TO nf
140 READ x$, y$
150 IF LEN x$ > 255 OR LEN y$ > 255 THEN SOUND .2,15: SOUND
    .3,16: PRINT "A função na instrução DATA tem mais que
    255 caracteres": STOP
160 LET cf = cf + LEN x$ + LEN y$ + 2
170 IF cf > = 1000 THEN SOUND .2,12: SOUND .3,13:
    SOUND .4,24: PRINT,, "Não existe espaço suficiente para
    a função - aumente o f na linha 30": STOP
180 LET f$(fant+1 TO cf) = CHR$ LEN x$ + x$+CHR$ LEN y$+y$
190 LET fant = cf
200 NEXT j
210 LET x$="" : LET y$=""
220 LET n=28: CLS: PRINT TAB 10; "ESPIRAIS"
230 PRINT ,,, TAB 14; "MENU"
240 PRINT,, "Introduzir uma função"; TAB n; "I"
250 PRINT,, "Desenhar uma função"; TAB n; "D"
260 PRINT,, "Armazenar uma função"; TAB n; "A"
270 PRINT,, "Parar o programa"; TAB n; "P"
280 INPUT r$
290 IF r$ <> "I" OR r$ <> "D" OR r$ <> "A" OR r$ <> "P"
    THEN GOTO 240
300 GOTO 300+30*(r$ ="I") + 130 * (r$ = "D") + 330*(r$="A") +
    410 * (r$ = "P")
310 INPUT "Pressione uma tecla para continuar"; z$
320 GOTO 220

```



```

330 PRINT,, TAB 7; "Introduza a função"
340 INPUT "Introduza a equação paramétrica x = f(t) para o
    eixo Ox"; x$
350 IF LEN x$ > 255 THEN SOUND .2,23: SOUND .3,24: PRINT,
    "A equação deve ter menos do que 255 caracteres.
    Entre com outra.": GOTO 340
360 PRINT ,,, "A equação é:", x$
370 INPUT "Introduza a equação paramétrica y = f(t) para o
    eixo Oy"; y$
380 IF LEN y$ > 255 THEN SOUND .1,23: SOUND .2,24 :
    PRINT,, "A equação deve ter menos do que 255 caracteres.
    Entre com outra.": GOTO 370
400 PRINT,,,"A equação é:", y$
410 GOSUB 1500
420 GOTO 310
430 PRINT,, "Aí vai se desenhar uma das funções já existentes no programa"
440 LET f = 0: LET p = 1
450 GOSUB 1800
460 IF s > = cf THEN GOTO 520
470 PRINT,, p; TAB 6; f$(s TO f); TAB 6;
480 GOSUB 1800
490 PRINT f$(s TO f)
500 LET p = p+1.
510 GOTO 450
520 INPUT "Entre com o número da função □"; q
530 IF q > p OR q < 0 THEN SOUND .2,23: SOUND .3,24: GOTO 520
540 LET f = 0: LET p=1
550 GOSUB 1800
560 IF p=q THEN LET x$ = f$ (s TO f)
570 GOSUB 1800
580 IF p <> q THEN LET p = p+1: GOTO 550
590 LET y$ = f$ (s TO f)
600 GOSUB 1500

```

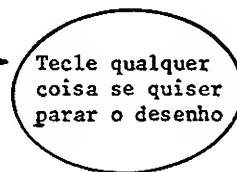


```

610 LET x$ = "": LET y$= ""
620 GOTO 310
630 PRINT,,; TAB.8; "Armazenar uma função"
640 IF LEN x$ = 0 OR LEN y$ = 0 THEN SOUND .2,23: SOUND .3,23
    : PRINT,, "A função não está definida": GOTO 310
650 LET cf = cf + LEN x$ + LEN y$ + 2
660 IF cf > = 1000 THEN SOUND .3,23: SOUND .2,24: PRINT,
    "Não existe espaço suficiente para guardar a função":
    LET cf = cf - LEN x$ - LEN y$ - 2: GOTO 310
670 LET f$ (fant + 1 TO cf) = CHR$ LEN x$ + x$ + CHR$
    LEN.y$ .+ y$
680 LET fant = cf
690 PRINT,, "A função já está armazenada"
700 GOTO 310
710 PRINT "Como você já cansou..."  


720 FOR k=1 TO 18
730 SOUND k/10, k+3
740 NEXT k
750 FOR k=18 TO 1 STEP -1
760 SOUND k/10,-(k+3)
770 NEXT k
780 PRINT "Vamos parar este programa, viu?!?!"  


790 STOP
1500 CLS: PRINT TAB 9; "ESPIRAIS"
1510 PRINT,, x$: PRINT y$  


1520 LET t=0: LET cx = 127: LET cy = 70
1530 LET rx = VAL x$ : LET ry = VAL y$  

1540 PLOT cx + rx, cy + ry
1550 LET t = t+.025
1560 IF INKEY$ <> "" THEN GOTO 1580 → Tecle qualquer
1570 GOTO 1530
1580 RETURN
1800 LET s = f+2
1810 LET f = s + CODE f$ (s-1) - 1
1820 RETURN

```

2000 PRINT ,,,,""
 2010 PRINT "Este programa permite representar na tela  
funções que foram introduzidas de forma independente  
para as coordenadas x e y "
 2020 PRINT "Ambas as equações devem ser expressas como  
função de t"
 2030 PRINT ,,,,"Neste programa tem-se a possibilidade de  
desenhar 6 gráficos diferentes"
 2040 PRINT ,,,; FLASH 1; "Aperte qualquer tecla para  
começar"
 2050 IF INKEY\$ = "" THEN GOTO 2050
 2060 RETURN  
 O.I. - Se as coisas estiverem ficando "pretas" quanto ao entendimento de algumas linhas ou quem sabe de todo o programa não esqueça que um meio para não ficar "boiando" é ter também o meu livro "BASIC no TK 90X". Ele o auxiliará muito para não ficar nas trevas...

# Capítulo 18

## Perspectiva cônica

Este programa permite ver de qualquer ângulo um sólido escolhido por você entre os que compõem o "menu": cubo, prisma hexagonal, pirâmide ou octaedro.

Utilizando as teclas do cursor pode-se fazer o sólido ir para a esquerda(tecla "5") para a direita (tecla "8"), aproximar o sólido (tecla "7") afastar o sólido (tecla "6"), subir o sólido (tecla "s") ou abaixar o sólido (tecla "b").

A tecla "m" permite voltar ao menu.

Aí está o "programão"

```
1 REM ** Perspectiva cônica **
2 BORDER 7
5 LET px = 0: LET py = 0: LET ang = PI: LET r = 200:
    LET c = 80: LET d = 50
10 PAPER 7: INK 0: BRIGHT 0
```

```
15 CLS
20 PRINT AT 3,6; "** PERSPECTIVA CÔNICA**"
25 PRINT AT 4,4; "Escolha um corpo da lista"
30 PRINT AT 5,5; "Tecle um dos números:"
32 PRINT AT 7,8; "1 - CUBO"
34 PRINT AT 8,8; "2 - PIRÂMIDE"
36 PRINT AT 9,8; "3 - PRISMA"
38 PRINT AT 10,8; "4 - OCTAEDRO"
39 PRINT AT 12,6; "*Durante o programa:"; AT 14,10; "7 =
aproximação"; AT 15,10; "6=afastamento"; AT 16,10; "5=
esquerda"; AT 17,10; "8=direita"; AT 18,10; "s=subir";
AT 19,10; "b=abaixar"; AT 20,10;"m=voltar ao menu"
40 IF INKEY$ = "1" THEN PRINT AT 2,6; "□ ESPERE □": PRINT
    AT 7,7; ">": GOTO 100
50 IF INKEY$ = "2" THEN PRINT AT 2,6; "□ ESPERE □": PRINT
    AT 8,7; ">": GOTO 250
60 IF INKEY$ = "3" THEN PRINT AT 2,6; "□ ESPERE □": PRINT
    AT 9,7; ">": GOTO 400
70 IF INKEY$ = "4" THEN PRINT AT 2,6; "□ ESPERE □": PRINT
    AT 10,7; ">": GOTO 550
90 GOTO 40
100 RESTORE 125: LET pt = 10: LET lin = 190
110 DIM x(pt): FOR t=1 TO pt:115 READ x(t)
120 NEXT t
125 DATA -50,-50,50,50,-50,-50,50,50,0,0
130 DIM y(pt): FOR t=1 TO pt
135 READ y(t)
140 NEXT t
145 DATA -50,50,50,-50,-50,50,50,-50,150,150
150 DIM z(pt): FOR t=1 TO pt
155 READ z(t)
160 NEXT t
165 DATA 0,0,0,0,100,100,100,100,0,125
170 PAPER 0: INK 7: BRIGHT 1
```



```

180 GOTO 1000
190 CLS
195 PLOT Ø,c: DRAW INK 5; BRIGHT Ø;255,Ø
200 RESTORE 225: FOR t=1 TO 13
210 READ v,w
215 PLOT 128+m(v), n(v): DRAW m(w)-m(v),n(w)-n(v)
220 NEXT t
225 DATA 1,2,2,3,3,4,4,1,1,5,2,6,3,7,4,8,8,5,5,6,6,7,7,
     8,9,1Ø
230 CIRCLE 128+m(1Ø),n(1Ø)+(n(1Ø)-n(9))/1Ø,
     (n(1Ø)-n(9))/1Ø
240 GOTO 1100
250 RESTORE 27Ø: LET pt = 9: LET lin=32Ø; LET ang=2.
    7488936
255 DIM x(pt): FOR t=1 TO pt
260 READ x(t)
265 NEXT t
270 DATA -5Ø,-5Ø,5Ø,5Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
275 DIM y(pt): FOR t=1 TO pt
280 READ y(t)
285 NEXT t
290 DATA -5Ø,5Ø,5Ø,-5Ø,Ø,15Ø,15Ø,17Ø,15Ø
295 DIM z(pt): FOR t=1 TO pt
300 READ z(t)
305 NEXT t
310 DATA Ø,Ø,Ø,Ø,1ØØ,Ø,5Ø,4Ø,3Ø
313 PAPER Ø: INK 7: BRIGHT 1
315 GOTO 1000
320 CLS
325 PLOT Ø,c: DRAW INK 5; BRIGHT Ø; 255,Ø
330 RESTORE 37Ø: FOR t=1 TO 11
340 READ v,w
350 PLOT 128+m(v),n(v): DRAW m(w)-m(v),n(w)-n(v)
360 NEXT t

```



```

37Ø DATA 1,2,2,3,3,4,4,1,1,5,2,5,3,5,4,5,6,7,7,8,8,9
38Ø GOTO 11ØØ
40Ø RESTORE 42Ø: LET pt = 12: LET lin=475: LET ang=PI/2:
     LET c=1ØØ
405 DIM x(pt): FOR t=1 TO pt
410 READ x(t)
415 NEXT t
42Ø DATA -1ØØ,-1ØØ,-1ØØ,-1ØØ,-1ØØ,1ØØ,1ØØ,1ØØ,1ØØ,1ØØ
425 DIM y(pt): FOR t=1 TO pt
430 READ y(t)
435 NEXT t
44Ø DATA 25,5Ø,25,-25,-5Ø,-25,25,5Ø,25,-25,-5Ø,-25
445 DIM z(pt): FOR t=1 TO pt
450 READ z(t)
455 NEXT t
46Ø DATA Ø,43.3,86.6,86.6,43.3,Ø,Ø,43.3,86.6,86.6,43.3,Ø
465 PAPER Ø: INK 7: BRIGHT 1
47Ø GOTO 1000
475 CLS
480 PLOT Ø,c: DRAW INK 5; BRIGHT Ø;255,Ø
485 RESTORE 5ØØ: FOR t=1 TO 18
490 READ v,w
495 PLOT 128+m(v),n(v): DRAW m(w)-m(v),n(w)-n(v)
50Ø NEXT t
505 DATA 1,2,2,3,3,4,4,5,5,6,6,1,7,8,8,9,9,1Ø,1Ø,11,11,12,12,
     7,1,7,2,8,3,9,4,1Ø,5,11,6,12
51Ø GOTO 11ØØ
55Ø RESTORE 57Ø: LET pt = 1Ø: LET lin=625: LET c = 1ØØ
555 DIM x(pt): FOR t=1 TO pt
560 READ x(t)
565 NEXT t
57Ø DATA -5Ø,-5Ø,5Ø,5Ø,Ø,Ø,-15,Ø,15,Ø
575 DIM y(pt): FOR t=1 TO pt
58Ø READ y(t)

```



```

585 NEXT t
590 DATA -50,50,50,-50,0,0,125,150,125,133,66
595 DIM 2(pt): FOR t=1 TO pt
600 READ z(t)
605 NEXT t
610 DATA 75,75,75,75,150,0,0,0,0,75
615 PAPER 0: INK 7: BRIGHT 1
620 GOTO 1000

```



```

625 CLS
630 PLOT 0,c: DRAW INK 5: BRIGHT 0;255,0
635 RESTORE 655: FOR t=1 TO 18

```

```

640 READ v,w
645 PLOT 128+m(v),n(v): DRAW m(w)-m(v),n(w)-n(v)
650 NEXT t

```

```

655 DATA 1,2,2,3,3,4,4,1,1,5,2,5,3,5,4,5,1,6,2,6,3,6,4,6,7,
     8,8,9,9,7,7,10,8,10,9,10
660 GOTO 1100

```

```

1000 LET a=r*COS ang: LET b=r*SIN ang: DIM m(pt): FOR t=1
    TO pt
1005 IF a=px THEN LET m(t)=d*((x(t)-a)/(y(t)-b)): GOTO 1030
1007 IF a=x(t) AND x(t) <> px THEN LET r=r+20: GOTO 1000
1010 LET m(t)=d*TAN(ATN((py-b)/(px-a))-ATN
    ((y(t)-b)/(x(t)-a)))

```

```

1020 IF 128+m(t) > 255 OR 128+m(t) < 0 THEN PRINT AT 1,5;
    "DEMASIADO PRÓXIMO": GOTO 1100

```

```

1030 NEXT t
1050 DIM n(pt): FOR t=1 TO pt
1060 LET n(t)=z(t)+((c-z(t))/SQR((a-x(t))*(a-x(t)) +
    (b-y(t))*(b-y(t))))*(SQR((a-x(t))*(a-x(t))+
    (b-y(t))*(b-y(t)))-SQR(m(t)*m(t)+d^2))

```

```

1070 IF n(t) > 175 OR n(t) < 0 THEN PRINT AT 1,5;
    "DEMASIADO PRÓXIMO": GOTO 1100

```

```

1080 NEXT t

```

```

1090 GOTO lin

```

```

1100 IF INKEY$ = "8" THEN LET ang = ang+PI/16: GOTO 1000

```

```

1110 IF INKEY$ = "5" THEN LET ang = ang-PI/16: GOTO 1000
1120 IF INKEY$ = "6" THEN LET r=r+20: GOTO 1000
1130 IF INKEY$ = "7" THEN LET r=r-20: GOTO 1000
1140 IF INKEY$ = "s" AND c+20 <= 175 THEN LET c=c+20: GOTO 1000
1150 IF INKEY$ = "b" AND c-20 >= 0 THEN LET c=c-20: GOTO 1000
1160 IF INKEY$ = "m" THEN PAPER 7: INK 0: BRIGHT 0: GOTO 5
1170 GOTO 1100

```

Execute (RUN) o programa e a sua "primeira" tela será do seguinte tipo

**\*\*PERSPECTIVA CÔNICA\*\***

Escolha um corpo da lista  
Tecle um dos números:

1	- CUBO
2	- PIRÂMIDE
3	- PRISMA
4	- OCTAEDRO

\*Durante o programa:

7	- aproximação
6	- afastamento
5	- esquerda
8	- direita
s	- subir
b	- abaixar
m	- voltar ao menu

Figura 25

No decorrer do uso do programa "PERSPECTIVA CÔNICA" podem surgir as seguintes situações na tela, (Figuras 26 a,b,c,d,e,f) onde se pode perceber facilmente a aproximação e o afastamento.

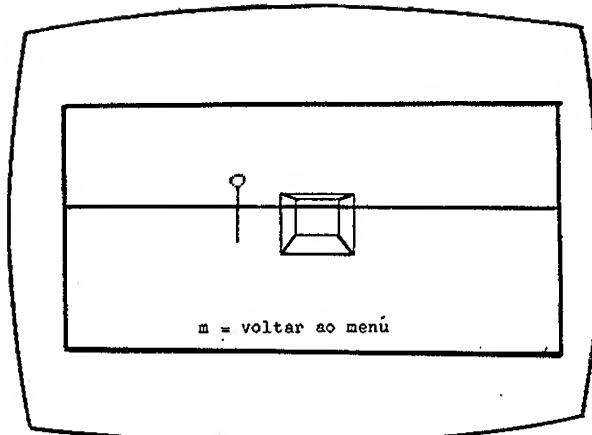


Figura  
26a

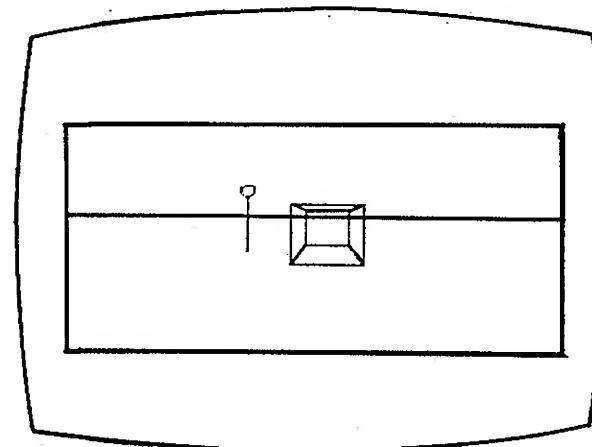


Figura  
26b

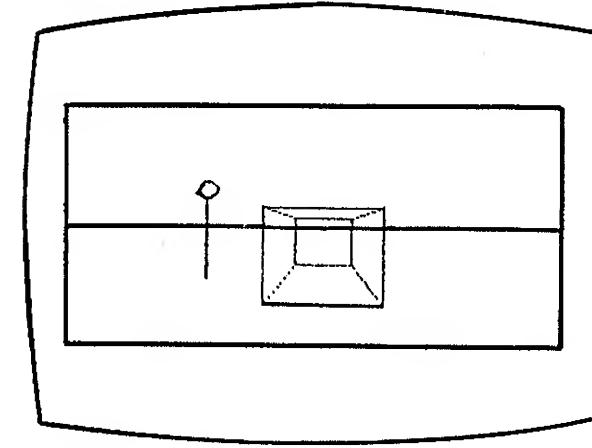


Figura  
26c

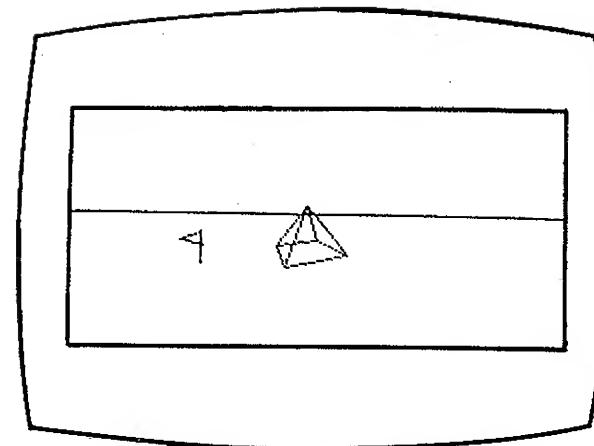


Figura  
26d

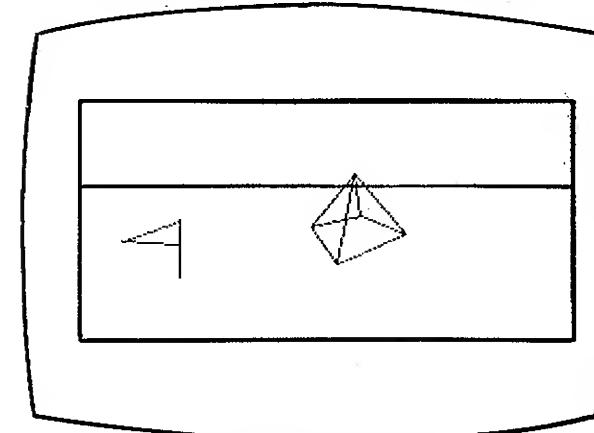


Figura  
26e

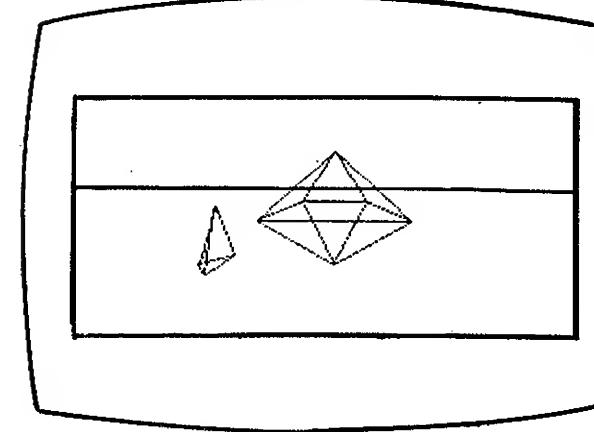


Figura  
26f

# Capítulo 19

## Uma aplicação para a Eletrônica com direito a uma distração-tipo destreza no circuito

Aí vai um outro "programão" porém agora voltado para a Eletrônica!

Neste caso específico vai se calcular a "etapa" amplificadora de um emissor comum.

Para esta finalidade devem ser introduzidos três dados: em primeiro lugar a tensão com a qual se deseja alimentar o circuito, em seguida a relação entre a tensão de entrada e de saída e finalmente a corrente do coletor do transistor, a qual depende do tipo de transistor que vai ser utilizado sendo que valores normais para a intensidade da corrente são em miliamperes (mA) ou em frações de ampere (A).

O programa é composto de quatro etapas básicas:

- 1) cálculo das componentes
- 2) geração do símbolo de resistência elétrica ou seja do ohm ( $\Omega$ )

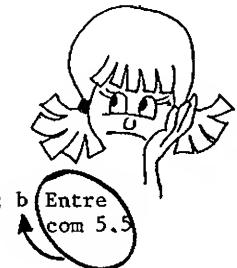
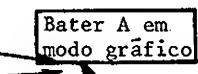
- 3) exibição da tabela das resistências comumente encontradas no comércio
- 4) exibição do esquema do circuito

Espero realmente que este programa seja um útil modelo para que os professores de Eletrônica possam elaborar outros esquemas semelhantes que permitam principalmente aos seus alunos um aprendizado individual tendo como professor o TK 90X.

```

10 REM ** Etapa amplificadora**
15 CLS: GOSUB 500: GOSUB 600
20 REM ** Cálculo de componentes **
30 INPUT "Tensão Vcc(V) = "; a
40 INPUT "Relação de amplificação Av = "; b
50 INPUT "Corrente do coletor Ic(A) = "; c
55 PRINT "Ic="; c; "A"
60 PRINT "Re >"; .7/c; "faça a escolha segundo a tabela"
65 GOSUB 190
70 INPUT "Re(A) ="; d: CLS
80 LET e=d*b: LET f=(.7+(d*c))
85 PRINT "Rc="; d*b; "A", "Re="; d; "A"
90 PRINT "Rl="; INT(e*a/f); "A", "R2="; (e/(1-(f/a))); "A"
110 PRINT "C1=C2=470 micro F/16V", "Ce="; INT(1.59/(20*d)*
    1000000); "microF"
120 PRINT "Pot.Tr1="; (a-(d*b+d)*c)*c*1000; "mW"
122 PRINT AT 19,0;"Vcc="; a;"V"; AT 19,15;"Ic="; c;"A"
125 GOSUB 600
130 PRINT AT 21,0; "Quer novo cálculo (S/N)?"
    IF INKEY$="" THEN GOTO 130
140 IF INKEY$="S" OR INKEY$="s" THEN RUN
150 STOP

```

```

185 REM **Aí vai a tabela das resistências encontradas no
comércio**
190 RESTORE 240
195 PRINT AT 4,2; "Ω"; AT 4,5; "Ω"; AT 4,9; "Ω"; AT 4,12;
"KΩ"; AT 4,16; "KΩ"; AT 4,19; "KΩ"; AT 4,23; "MΩ"; AT
4,27; "MΩ"
200 FOR l = 6 TO 17: READ g
210 PRINT AT l,1;g;AT l,5;g*10; AT l,8;g*100; AT l,12;g;
AT l,16;g*10; AT l,19;g*100; AT l,23;g
220 IF l <= 10 THEN PRINT AT l,27;g*10
230 NEXT l
240 DATA 1,1.2,1.5,1.8,2.2,2.7,3.3,3.9,4.7,5.6,6.8,8.2
250 RETURN
500 REM *Gráfico Ω*
505 RESTORE 510: FOR k=0 TO 7: POKE USR "A"+k,j:
NEXT k
510 DATA 24,36,66,129,129,66,36,231
520 RETURN
600 REN **Aí vai o desenho do circuito**
605 RESTORE 640: PRINT "Cálculos para o circuito"
610 FOR k=1 TO 49: READ p,q,w,z
620 PLOT p,q: DRAW w,z
630 NEXT k
640 DATA 74,74,0,7,74,81,9,0,83,84,0,-8,87,84,0,-8,87,81,27,
0,114,85,0,-8,100,81,3,0,97,89,6,0,103,39,0,8,103,97,-6,
0,97,97,0,-8
650 DATA 100,105,0,-3,100,105,55,0,100,89,0,-8,126,105,0,-8,
123,97,6,0,129,97,0,-8,129,89,-6,0,123,89,0,8,126,89,0,
-5
660 DATA 126,34,-4,0,122,84,-6,-3,74,66,0,-7,74,59,79,0,100,
59,0,7,97,66,6,0,103,66,0,8,103,74,-6,0,97,74,0,-8,100,
74,0,7

```

Para obter o  $\Omega$  bata A em modo gráfico

```

670 DATA 126,59,0,6,123,66,0,129,66,0,8,129,74,-6,0,123,
74,0,-8,126,74,0,5,126,79,-5,0,121,79,-6,3
680 126,84,10,0,136,87,0,-6,139,87,0,-6,139,84,7,0,146,84,
0,-7,146,59,0,7
690 DATA 126,79,10,0,136,79,0,-7,133,72,6,0,133,69,6,0,136,
69,0,-10
700 CIRCLE 115,82,7
710 PRINT AT 13,5; "ENT."; AT 10,9; "C1"; AT 7,12; "R1";
AT 13,10; "R2"; AT 13,13; "Re"; AT 13,19; "SAI."
720 PRINT AT 15,16; "Ce"; AT 7,15; "Rc"; AT 10,17; "C2";
AT 8,20; "+V"; AT 14,20;"-V"
730 RETURN

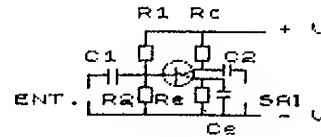
```

Ao executar (RUN) este programa você deve ver a seguinte sequência:



a)

### Cálculos para o circuito



b)

Resposta: Faça a escolha segundo a tabela

$\Omega$	$\Omega$	$\Omega$	$K\Omega$	$K\Omega$	$K\Omega$	$M\Omega$	$M\Omega$
1	10	100	1	10	100	1	10
2	20	200	2	20	200	2	20
3	30	300	3	30	300	3	30
4	40	400	4	40	400	4	40
5	50	500	5	50	500	5	50
6	60	600	6	60	600	6	60
7	70	700	7	70	700	7	70
8	80	800	8	80	800	8	80
9	90	900	9	90	900	9	90
10	100	1000	10	100	1000	10	100

Figura 27

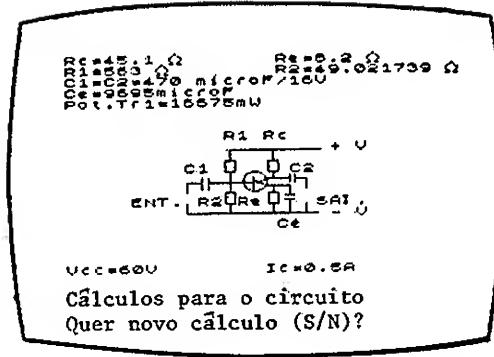


Figura 27c

Se por acaso de circuito eletrônico o seu conhecimento é nulo, e o interesse pelo assunto há pouco apresentado foi nenhum quero remediá-lo(a) com uma distração.

Aí vai o jogo o "CARRO MALUCO"...

Neste jogo você deve dirigir um carro que "trafega" a toda velocidade num dos circuitos apresentados nas Figura 28 a,b ou c porém com o detalhe de estar sem freios.

Para conduzir o carro você pode utilizar as teclas "P" e "O". A tecla "P" faz o carro se movimentar no sentido horário e com a tecla "O" se desloca o carro no sentido anti-horário.

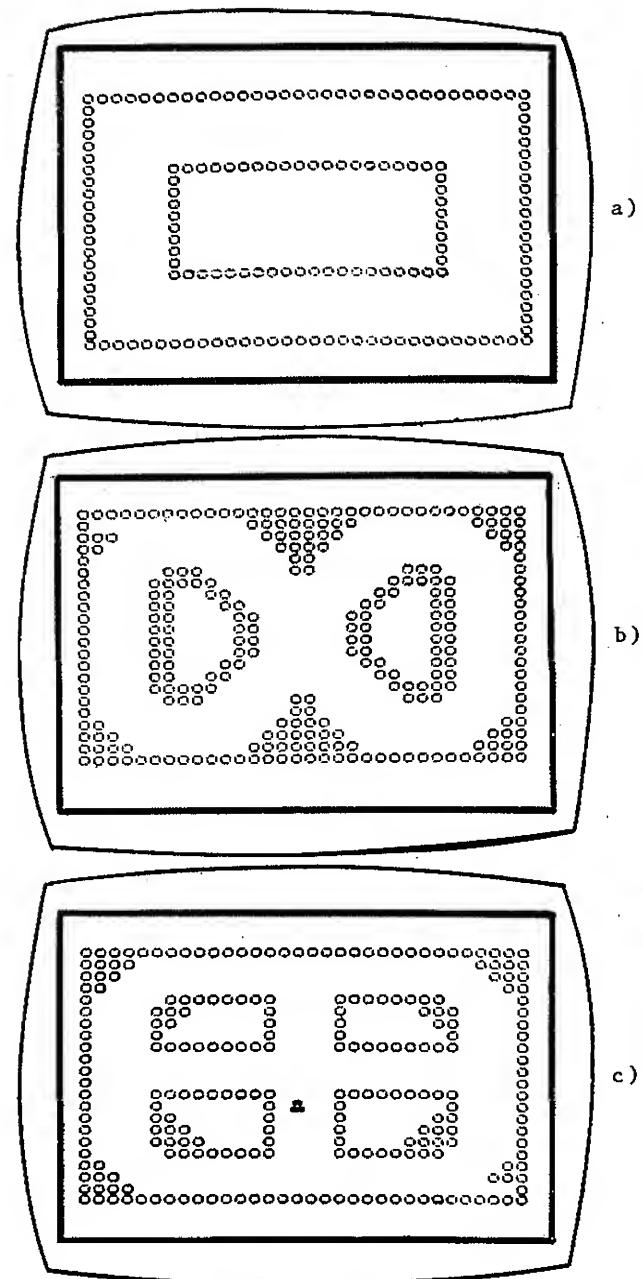


Figura 28

Como já disse o jogo consta de três circuitos sobre os quais você deve permanecer durante 20 segundos ainda que o seu TK 90X represente este tempo como sendo 1 minuto.

No início do jogo o TK 90X lhe pedirá o seu nível de destreza (ou de competência) o qual pode variar de 1 até 9.

Tenha muito cuidado com as derrapagens pois quanto mais alto for o nível escolhido por você maior será a possibilidade que o seu carro derrapse.

Bem aí vai o "programão" e esteja atento(a)...

```

5 REM ** CARRO MALUCO **
10 GOSUB 8000
20 REM** Rotina principal **
30 FOR c=1 TO 3
40 GOSUB 8500
50 BRIGHT 0: FLASH 0: BORDER 4: PAPER 4: INK 0: CLS
60 GOSUB c*1000
70 DIM a(4)
80 LET ang=1
90 LET q=1: LET w=1
100 FOR t=1 TO 180: REM + TEMPO *
110 PRINT AT q,w;CHR$ (143+ang)
120 IF INKEY$="o" OR INKEY$="O" THEN LET ang=ang-1: IF
    ang=0 THEN LET ang=4
130 IF INKEY$="p" OR INKEY$="P" THEN LET ang=ang+1: IF
    ang=5 THEN LET ang=1
140 LET a(ang)=niv
150 LET o=q: LET p=w
160 IF a(1) > 0 THEN LET p=p+1: LET a(1)=a(1)-1: IF
    a(1) < niv-1 THEN SOUND .018,24: SOUND .018,23

```



```

170 IF a(2) > 0 THEN LET o=o+1: LET a(2)=a(2)-1:
    IF a(2) < niv-1 THEN SOUND .018,24: SOUND .018,23
180 IF a(3) > 0 THEN LET p=p-1: LET a(3)=a(3)-1:
    IF a(3) < niv-1 THEN SOUND .018,24: SOUND .018,23
190 IF a(4) > 0 THEN LET o=o-1: LET a(4)=a(4)-1:
    IF a(4) < niv-1 THEN SOUND .018,24: SOUND .018,23
200 PRINT AT q,w;" "
210 IF ATTR (o,p)=39 THEN GOTO 7000
220 LET q=o: LET w=p
230 NEXT t
240 NEXT c
245 REM ** FINAL FELIZ **
250 LET c$="oooooooooooooooooooo"
260 FOR m=1 TO 21: LET c$=c$+c$ (1 TO 32): NEXT m
270 FOR m=7 TO 0 STEP -1
280 PRINT AT 0,0; INK m; OVER 1; c$
290 SOUND .5,m*3: NEXT m
310 LET z$="Parabéns
    Você conseguiu dominar o carro. Queres tentar outra vez (S/N)? O seu nível de destreza passa a ser 1 unidade maior.
    PR
    EPARA-TE!!!"
320 BORDER 0: PAPER 1: CLS: INK 6: BRIGHT 1
330 PRINT AT 0,0;
350 FOR m=1 TO 220
360 PRINT z$(m); FLASH 1; "_" ;CHR$ 8;
365 SOUND .05,24: SOUND .06,23
370 NEXT m
380 IF INKEY$="" THEN GOTO 380
390 IF INKEY$="S" OR INKEY$="s" THEN GOTO 420
400 IF INKEY$="n" OR INKEY$="N" THEN GOTO 9900
410 GOTO 310

```



Bata exatamente assim!!!

```

420 FOR m=221 TO LEN z$
430 PRINT z$(m); FLASH 1; "_" ; CHR$ 8;
440 SOUND .05,20: SOUND .06,21
450 NEXT m
460 LET niv=niv+1
470 SOUND 1,10: SOUND 1,12
480 GOTO 30
1000 REM ** CIRCUITO 1 **
1010 PRINT AT 0,0; INK 7; "00000000000000000000000000000000"
1020 PRINT AT 21,0; INK 7;"00000000000000000000000000000000"
1030 FOR a=0 TO 21
1040 PRINT AT a,0; INK 7; "0"; AT a,31; INK 7;"0"
1050 NEXT a
1060 PRINT AT 6,6; INK 7; "000000000000000000000000"
1070 PRINT AT 15,6;INK 7; "000000000000000000000000"
1080 FOR a=6 TO 15
1090 PRINT AT a,6; INK 7; "0"; AT a,25; INK 7,"0"
1100 NEXT a
1110 RETURN
2000 REM ** CIRCUITO 2 **
2005 PRINT AT 0,0; INK 7; "00000000000000000000000000000000"
    AT 21,0; "00000000000000000000000000000000"
2010 PRINT AT 1,0; INK 7; "0"; AT 1,28; "0000000";
    AT 2,29; "00000"; AT 3,30; "00"; AT 18,0; "00";
    AT 18,30; "00000"; AT 19,29; "0000000";
    AT 20,28; "0000"
2020 FOR m=4 TO 17
2030 PRINT AT m,0; INK 7; "0"; AT m,31; "0"
2040 IF m > 4 AND m < 10 OR m > 12 AND m < 17 THEN PRINT
    AT m,5; INK 7; "0"; AT m,13; "0"; AT m,18; "0";
    AT m,26;"0"
2050 NEXT m

```



```

2060 PRINT AT 4,6; INK 7; "00000000"; AT 4,18; "00000000";
    AT 5,6; "000"; AT 5,23; "000"; AT 6,6; "00"; AT 6,24;
    "00"; AT 7,6; "0"; AT 7,25; "0"; AT 9,5; "0000000000";
    AT 9,18; "0000000000".
2070 PRINT AT 13,5; INK 7; "0000000000"; AT 13,18;
    "0000000000"; AT 15,6; "0"; AT 15,25; "0"; AT 16,6;
    "00"; AT 16,24; "00"; AT 17,6; "00000000"; AT 17,18;
    "00000000"; AT 18,1; "0"; AT 18,30; "00000";
    AT 19,29; "0000000"; AT 20,28; "0000"
2080 RETURN
3000 REM ** CIRCUITO 3 **
3005 PRINT AT 0,0; INK 7; "00000000000000000000000000000000"
    AT 20,28; "00000000000000000000000000000000"
3010 FOR m=1 TO 20
3020 PRINT AT m,0; INK 7; "0"; AT m,31; "0"
3030 NEXT m
3040 PRINT INK 7; AT 1,12; "00000000"; AT 1,28; "0000000";
    AT 2,13; "000000"; AT 2,29; "00000"; AT 3,14; "0000";
    AT 3,30; "00"; AT 4,15; "00"; AT 5,6; "000"; AT 5,15;
    "00"; AT 5,23; "000"; AT 6,5; "00000"; AT 6,22;
    "00000"; AT 7,5; "00□□00"; AT 7,21; "00□□00";
    AT 8,5; "00□□00"; AT 8,20; "00□□00"; AT 9,5;
    "00□□□□00"; AT 9,19; "00□□□□00"; AT 10,5; "00□□□□00";
    AT 10,19; "00□□□□00"
3050 PRINT INK 7; AT 11,5; "00□□□□00"; AT 11,19; "00□□□□00";
    AT 12,5; "00□□□□00"; AT 12,19; "00□□□□00"; AT 13,5;
    "00□□□□00"; AT 13,20; "00□□□□00"; AT 14,5; "00□□00";
    AT 14,21; "00□□00"; AT 15,5; "00000"; AT 15,22;
    "00000"; AT 16,6; "000"; AT 16,15; "00"; AT 16,23;
    "000"; AT 17,15; "00"; AT 18,1; "0"; AT 18,14; "0000";
    AT 18,30; "00000"; AT 19,13; "000000"; AT 19,29;
    "0000000"; AT 20,12; "00000000"
3060 RETURN

```

```

7000 REM ** DESFECHE FATAL **
7005 PRINT AT 0,p; PAPER 7; INK 1; CHR$ (ang+143):
    SOUND .5,-10
7010 PRINT AT 21,0; INK 7; PAPER 0; "Circuito";c,"Tempo";
    INT(t/3)
7020 PRINT #1; "Outra (S/N)?"
7030 IF INKEY$="n" OR INKEY$="N" THEN GOTO 9000
7040 IF INKEY$="s" OR INKEY$="S" THEN RUN
7050 GOTO 7030
8000 REM ** APRESENTAÇÃO **
8005 RESTORE 8000 :FOR a=1 TO 4
8010 READ a$
8020 FOR j=0 TO 7
8030 READ c: POKE USR a$+j,c
8040 NEXT j
8050 NEXT a
8060 DATA "a",224,230,255,231,231,255,230,224,"b",255,255,
      255,36,36,126,126,60,"c",7,103,255,231,231,255,103,7,
      "d",60,126,126,36,36,255,255,255
8070 PAPER 2: BORDER 2: INK 7: CLS
8080 FOR a=0 TO 21 STEP 2
8090 PRINT AT a,0; PAPER 6; INK 3; FLASH 1; "carro maluco"
8100 NEXT a
8110 FOR a=1 TO 21 STEP 2
8120 PRINT AT a,0; PAPER 3; INK 6; FLASH 1; "carro maluco"
8130 NEXT a
8140 PRINT AT 0,15; FLASH 1; "Cuidado..."
8150 PRINT AT 2,15; "O seu veloz carro"; AT 3,15; "está sem
      freios"; AT 4,15; "no circuito."; AT 10,15; "P=horário";
      AT 12,15; "0=anti-horário"
8160 PRINT #1; "Escolha o nível (1-9)"
8170 LET m$=INKEY$
8180 IF CODE m$ > 48 AND CODE M$ < 58 THEN GOTO 8200

```



```

8190 GOTO 8170
8200 LET niv=VAL M$
8210 RETURN
8500 REM ** NÚMERO DE CIRCUITO **
8505 PAPER 0: BORDER 4: CLS
8510 LET b$="circuito "+STR$ c+" "
8520 FOR a=1 TO 51
8530 IF A+31 >= 52 THEN PRINT AT 11,0; PAPER 1; INK 6;
      B$(A TO 51) : GOTO 8550
8540 PRINT AT 11,0; PAPER 1; INK 6; B$(a TO A+31)
8550 SOUND ,05,a/2
8560 NEXT a
8570 RETURN
9000 REM ***** FIM ****
9005 BRIGHT 0: OVER.0: FLASH 0: INVERSE 0: PAPER 7: INK 0:
      BORDER 7 : SOUND 1,0: CLS

```

Ao executar o programa deve surgir a seguinte apresentação na tela:

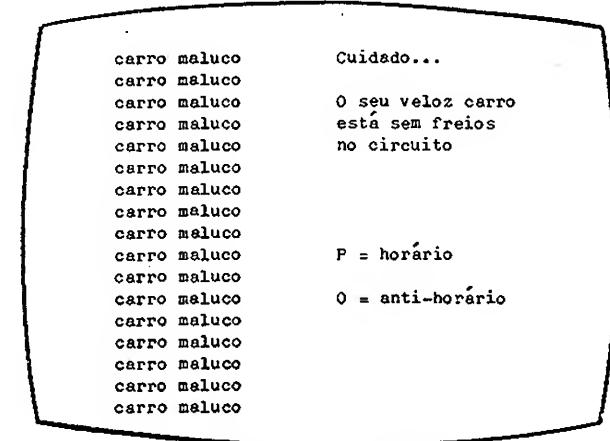


Figura 29



# Capítulo 20

## Simulador lógico além de um super jogo

Chegamos ao último capítulo e nele apresento dois interessantíssimos programas (acho que não é pretensão da minha parte...).

No primeiro programa você terá a oportunidade de "refinar" os seus conhecimentos de Eletrônica (uai outra vez?!?!) naquilo que os(as) "entendidos"(“as”) chamam de portas lógicas.

Acho que o tema é bastante didático sem deixar de ser divertido, visto ser este um ponto que não quero esquecer nunca...

O programa "SIMULADOR LÓGICO" ilustra o funcionamento das portas lógicas usuais nas quais aliás se inspira toda a tecnologia atual, inclusive do seu maravilhoso TK 90X.

Você já está mais interessado(a), não é?

E é precisamente com o auxílio do TK 90X que você estimado(a) leitor(leitora) aprenderá como

se comportam estas portas diante da informação de "zeros" e "uns" (ou seja sem tensão ou com tensão) que se aplicam as entradas. Dependendo dos valores destes e do tipo da porta obter-se-á um determinado valor de saída.

No programa aparecerá o esquema elétrico equivalente à função lógica em forma de interruptores, resistências e lâmpadas, incluindo-se o símbolo eletrônico que representa a porta com os valores nas suas entradas e na saída.

Em cada esquema se representam as entradas com interruptores indicando-se com "0" quando estão abertas e com "1" quando estão fechadas.

Os sinais "+" e "-" representam os polos de uma pilha e as resistências são representadas por retângulos.

O TK 90X lhe pedirá que introduza os valores que você quer ter nas entradas e depois lhe exibirá o valor na saída.

Aí está o programa "SIMULADOR LÓGICO"

```

10 REM *** SIMULADOR LÓGICO ***
20 INK 0: BORDER 7: PAPER 7
30 OVER 0: BRIGHT 0: FLASH 0: CLS
40 REM GRÁFICOS DEFINIDOS
50 LET z=99: LET x=192: LET y=255
60 LET q=0: LET v=28: LET w=8
70 DATA y,y,x,x,x,y,y,q,y,y,q,q,d,y,y,q
80 DATA y,y,3,3,3,y,y,q,127,127,z,z,z,z,z,z
90 DATA z,z,z,z,z,z,z,z,z,z,z,z,z,z,z,z,z,z
100 DATA 24,126,126,y,y,126,126,24,v,v,1,2,4,w,v,v
110 DATA 4,w,211,227,195,q,q,q,v,v,w,w,w,w,v,v

```

```

120 DATA q,q,195,y,195,q,q,q
130 FOR n=USR "a" TO USR "a"+87: READ a: POKE n,a: NEXT n
140 REM menu principal
150 PRINT AT 1,8; "SIMULADOR LÓGICO"
160 PLOT 63,158: DRAW 129,0
170 PLOT 63,156: DRAW 129,0
180 PLOT 15,144: DRAW 225,0
190 DRAW 0,-128: DRAW -225,0: DRAW 0,128
200 PLOT 13,146: DRAW 229,0
210 DRAW 0,-132: DRAW -229,0: DRAW 0,132
220 PRINT AT 5,10; "escolha a opção"
230 PLOT 79,126: DRAW 96,0
240 LET a$="□= porta tipo □"
250 PRINT AT 7,4; "A "+a$+"INVERSOR"
260 PRINT AT 9,4; "B "+a$+"AND"
270 PRINT AT 11,4; "C "+a$+"OR"
280 PRINT AT 13,4; "D "+a$+"NAND"
290 PRINT AT 15,4; "E "+a$+"NOR"
300 PRINT AT 17,4; "F "+a$(TO 13) + "OR EXCLUSIVO"
310 PAUSE 0
320 GOSUB 2210
330 REM *** Escolha das portas
340 LET a=PEEK 23560
350 IF a=65 OR a=97 THEN GOTO 0440
360 IF a=66 OR a=93 THEN GOTO 0660
370 IF a=67 OR a=99 THEN GOTO 0950
380 IF a=68 OR a=100 THEN GOTO 1260
390 IF a=69 OR a=101 THEN GOTO 1560
400 IF a=70 OR a=102 THEN GOTO 1870
410 PRINT AT 11,3; "□ Pressione A,B,C,D,E,F": SOUND 3,13
415 SOUND 4,14: SOUND 4,15
420 PRINT AT 11,4; a$(TO 23)
430 GOTO 220

```



A,B,C,D,E,F  
em modo gráfico

```

440 REM ** INVERSOR ***
460 REM *** Aqui desenha-se o esquema ***
480 PRINT AT 18,6; "PORTA TIPO INVERSOR": GOSUB 2180
490 PLOT AT 10,19; __□□__
500 CIRCLE 194,88,2: OVER 1: PLOT 193,88: PLOT 194,88:
      PLOT 195,88: OVER 0
510 PLOT 192,88: DRAW -24,-16: DRAW 0,32: DRAW 24,-16
520 PRINT AT 8,19; "A□□□□S"
530 PRINT AT 8,3; "+□□ABC": PLOT 39,108: DRAW 8,0
540 PLOT 71,108: DRAW 37,0: DRAW 0,-12 → ABC
      em modo gráfico
550 PLOT 108,37: DRAW 0,-12: DRAW -67,0
560 PLOT 91,108: DRAW 0,-12:
570 PLOT 91,87: DRAW 0,-12
580 PRINT AT 10,11; "■□■"; AT 12,3;"-"
590 PRINT AT 11,10; "s"; AT 11,14; "a"
600 REM ** controle inversor **
610 PRINT AT 21,0; "Tecle o valor de A 1/0 M=menu ":
      PAUSE 0
620 IF INKEY$="0" THEN PRINT AT 10,13; "H"; AT 13,19;
      "000000"; INK 5; AT 10,11; "G": SOUND .3,10
630 IF INKEY$="1" THEN PRINT AT 10,13; "J"; AT 13,19;
      "100000"; INK 1; AT 10,11; "G": SOUND .2,25
640 IF INKEY$="m" OR INKEY$="M" THEN GOSUB 2210: GOTO 150
650 GOTO 590
660 REM Porta AND → GHJ
      em modo gráfico
680 REM ** Aqui desenha-se o esquema **
690 GOSUB 2170
700 PRINT AT 9,19; " "; AT 11,19; " "; AT 10,24; " "
710 PLOT 167,103: DRAW 0,-30: DRAW 10,0: DRAW 0,30,3:
      DRAW -10,0
720 PRINT AT 8,3; "+□□□□■"; AT 12,3; "-"; AT 10,13;
      "■"; AT 13,8; "Porta tipo AND"
730 PLOT 39,108: DRAW 68,0: DRAW 0,-35: DRAW -68,0

```

```

740 PRINT AT 9,6; "a□□b"; AT 10,12; "s"
750 PRINT AT 8,19; "A"; AT 13,19; "B"; AT 9,25; "S"
760 REM ** Controle AND ***
770 PRINT AT 21,1; "tecle o valor de A 1/0 (M=menu)":
PAUSE Ø
780 PRINT AT 10,27;"□"; AT 11,17; "□"
790 IF INKEY$="m" OR INKEY$="M" THEN GOSUB 2210: GOTO 150
800 LET a$=INKEY$
810 IF a$ <> "1" AND a$ <> "Ø" THEN GOTO 790
820 LET a=VAL a$: PRINT AT 9,17; a: IF a=Ø THEN PRINT AT
8,7;"I ": GOTO 840
830 PRINT AT 8,7; " K" I,K,G  
em modo gráfico
840 SOUND .2,20
850 PRINT AT 21,1;"□tecle o valor de B 1/0 □□□□□□□□": PAUSE Ø
860 LET b$=INKEY$
870 IF b$ <> "1" AND b$ <> "Ø" THEN GOTO 860
880 LET b=VAL b$: PRINT AT 11,17;b: IF b=Ø THEN PRINT AT
8,10;"I ": GOTO 900
890 PRINT AT 8,10; "K"
900 LET s=a AND b: PRINT AT 10,27;s
910 SOUND .2,15: SOUND .3,16.
920 IF s=Ø THEN PRINT INK 1; AT 10,13; " G": GOTO 940
930 PRINT INK 5; AT 10,13; " G"
940 PAUSE 3: GOTO 760
950 REM ** Porta OR **
960 REM ** Aqui desenha-se o esquema **
980 GOSUB 2170
990 PRINT AT 9,19; "__"; AT 11,19; "__"; AT 10,24; "__"
1000 PLOT 167,103: DRAW Ø,-3Ø,-1.5: DRAW 1Ø,Ø: DRAW Ø,3Ø,
DRAW -1Ø,Ø
1010 PRINT AT 8,19; "A"; AT 13,19; "B"; AT 9,25; "S"
1020 PRINT AT 7,9; "a"; AT 11,9; "b"; AT 11,12; "s"

```

```

1030 PRINT AT 8,3; "+□□□□□□□"; AT 1Ø,1Ø; "■"; AT 1Ø,13;
"■"; AT 12,3; "__"
1040 PLOT 39,108: DRAW 6Ø,-35: DRAW Ø,-68,Ø
1050 PLOT 75,108: DRAW Ø,-16: DRAW 16,Ø: DRAW Ø,16
1060 PRINT AT 18,8; "PORTA TIPO OR"
1070 REM ** controle OR ***
1080 PRINT AT 21,1;"□ tecle o valor de A 1/0 M=menu":
PAUSE Ø
1090 PRINT AT 10,27; "□"; AT 11,17; "□"
1100 IF INKEY$="m" OR INKEY$="M" THEN GOSUB 2210: GOTO 150
1110 LET a$=INKEY$
1120 IF a$ <> "1" AND a$ <> "Ø" THEN GOTO 1110
1130 LET a=VAL a$: PRINT AT 9,17; a: IF a=Ø THEN PRINT AT
8,1Ø;" I": GOTO 1150
1140 PRINT AT 8,1Ø;" K" I,K,G  
em modo gráfico
1150 SOUND .2,25
1160 PRINT AT 21,1; "tecle o valor de B 1/0 □□□□□□□□": PAUSE Ø
1170 LET b$=INKEY$
1180 IF b$ <> "1" AND b$ <> "Ø" THEN GOTO 1160
1190 LET b=VAL b$: PRINT AT 11,17;b: IF b=Ø THEN PRINT AT
1Ø,1Ø;" I": GOTO 1210
1200 PRINT AT 10,1Ø;" K"
1210 SOUND .2,15
1220 LET s=a OR b: PRINT AT 10,27;s
1230 IF s=Ø THEN PRINT INK 1; AT 10,13; " G": GOTO 1250
1240 PRINT INK 5; AT 10,13; " G"
1250 PAUSE 3: GOTO 1080
1260 REM ** porta NAND **
1270 REM ** Aqui desenha-se o esquema **
1290 GOSUB 2170
1300 PRINT AT 9,19; "__"; AT 11,19; "__"; AT 10,24; "__"
1310 CIRCLE 194,88,2: OVER 1: PLOT 193,88: PLOT 194,88:
PLOT 195,88: OVER Ø

```

```

1320 PLOT 167,103: DRAW 0,-30: DRAW 10,0: DRAW 0,30,3:
    DRAW -10,0
1330 PRINT AT 8,19; "A"; AT 13,19; "B"; AT 9,25; "S"
1340 PLOT 39,108: DRAW 68,0: DRAW 0,-32: DRAW -68,0:
    PLOT 64,108: DRAW 0,-32
1350 PRINT AT 8,3; "+"; AT 8,6; "ABC"; AT 8,12; "■";
    AT 12,12; "■"; AT 12,3; "-"; AT 10,10; "■"
1360 PRINT AT 18,8; "PORTA TIPO NAND"; AT 7,13; "a";
    AT 13,13; "b"; AT 10,9;"s"           I,G,K ABC
1370 REM ** controle NAND ***          em modo gráfico
1380 PRINT AT 21,1;"□tecle o valor de A 1/0 M=menu":
    PAUSE 0
1390 PRINT AT 10,27;"□"; AT 11,17;"□"
1400 IF INKEY$="m" OR INKEY$="M" THEN GOSUB 2210: GOTO 150
1410 LET a$=INKEY$
1420 IF a$<>"1" AND a$<>"0" THEN GOTO 1410
1430 LET a=VAL a$: PRINT AT 9,17;a: IF a=0 THEN PRINT AT
    8,12;" I":GOTO 1450
1440 PRINT AT 8,12; " K"
1450 SOUND .2,20: SOUND .3,21
1460 PRINT AT 21,1; "tecle o valor de B 1/0□□□□□□□□□":
    PAUSE 0
1470 LET b$=INKEY$
1480 IF b$<>"1" AND b$<>"0" THEN GOTO 1460
1490 LET b=VAL b$: PRINT AT 11,17;b: IF b=0 THEN PRINT AT
    12,12;" I": GOTO 1510
1500 PRINT AT 12,12; " K"
1510 SOUND 2,10
1520 LET s=NOT (a AND b): PRINT AT 10,27;s
1530 IF s=0 THEN PRINT INK 1; AT 10,10; " G": GOTO 1550
1540 PRINT INK 5; AT 10,10; " G"
1550 PAUSE 3: GOTO 1380
1560 REM ** Porta NOR **

```

```

1570 REM ** Aqui desenha-se o esquema
1590 GOSUB 2170
1600 PRINT AT 9,19;"_"; AT 11,19;"_"; AT 10,24;"_"
1610 PLOT 167,103: DRAW 0,-30,-1.5: DRAW 10,0: DRAW 0,30,3:
    DRAW -10,0
1620 PRINT AT 8,19; "A"; AT 13,19; "B"; AT 9,25; "S"
1630 CIRCLE 194,83,2: OVER 1: PLOT 193,88: PLOT 194,88:
    PLOT 195,88: OVER 0
1640 PRINT AT 11,8; "a"; AT 11,10; "b"; AT 11,12; "s"
1650 PLOT 39,108: DRAW 68,0: DRAW 0,-32: DRAW -68,0: PLOT
    76,108: DRAW 0,-32: PLOT 92,108: DRAW 0,-32
1660 PRINT AT 3,3; "+"; AT 8,6; "ABC"; AT 10,11; "■";
    AT 10,13; "■"; AT 12,3; "-"; AT 10,9; "■"
1670 PRINT AT 13,8; "PORTA TIPO NOR"           ABC
                                                H,J
1680 REM ** controle NOR **          em modo gráfico
1690 PRINT AT 21,0;"□tecle o valor de A 1/0 M=menu":
    PAUSE 0
1700 PRINT AT 10,27; "□"; AT 11,17;"_"
1710 IF INKEY$="m" OR INKEY$="M" THEN GOSUB 2210: GOTO 150
1720 LET a$=INKEY$
1730 IF a$<>"1" AND a$<>"0" THEN GOTO 1720
1740 LET a=VAL a$: PRINT AT 9,17;a: IF a=0 THEN PRINT AT
    10,9; " H": GOTO 1760
1750 PRINT AT 10,9; "J"
1760 SOUND .2,25: SOUND .3,24
1770 PRINT AT 21,1; "tecle o valor de B 1/0□□□□□□□□□":
    PAUSE 0
1780 LET b$=INKEY$
1790 IF b$<>"1" AND b$<>"0" THEN GOTO 1780
1800 LET b=VAL b$: PRINT AT 11,17;b: IF b=0 THEN PRINT
    AT 10,11;" H": GOTO 1820.
1810 PRINT AT 10,11; " J"
1820 SOUND .2,15: SOUND .3,16

```

```

1830 LET s=NOT (a OR b): PRINT AT 10,27;s
1840 IF s=0 THEN PRINT INK 1; AT 10,13; " G": GOTO 1680
1850 PRINT INK 5; AT 10,13; "G" G,D,E,F,H,J
1860 PAUSE 3: GOTO 1680 em modo gráfico
1870 REM ** Porta OR exclusivo **
1880 REM ** Aqui desenha-se o esquema
1890 GOSUB 2170
1910 PRINT AT 9,19; "__"; AT 11,19;"__"; AT 10,24;"__"
1920 PLOT 167,103: DRAW 0,-30,-1.5: DRAW 10,0: DRAW 0,30,3:
      DRAW -10,0: PLOT 163,103: DRAW 0,-30,-1.5
1930 PRINT AT 8,19; "A"; AT 13,19; "B"; AT 9,25; "S"
1940 PRINT AT 8,3; "+"; AT 12,10;"■"; AT 13,6; "a■□□s□□b"
      AT 14,3; "-"
1950 PLOT 39,108: DRAW 69,0: DRAW 0,-50: DRAW -69,0: PLOT
      60,108: DRAW 0,-50: PLOT 60,76: DRAW 43,0
1960 PRINT AT 18,4; "Porta tipo OR exclusivo"
1970 PRINT AT 9,7;"D□□□□□□D";AT 10,7; "E□□□□ E"; AT
      11,7; "F□□□□ F"
1980 REM ** controle exclusivo **
1990 PRINT AT 21,0;"Tecle o valor de A 1/0 M=menu":
      PAUSE 0
2000 PRINT AT 10,27; "□"; AT 11,17; "□"
2010 IF INKEY$="m" OR INKEY$="M" THEN GOSUB 2210: GOTO 150
2020 LET a$=INKEY$
2030 IF a$ <> "1" AND a$ <> "0" THEN GOTO 2020
2040 LET a=VAL a$: PRINT AT 9,17; a: IF a=0 THEN PRINT
      AT 13,7; " H": GOTO 2060
2050 PRINT AT 13,7; " J"
2060 SOUND .2,24: SOUND .3,23
2070 PRINT AT 21,1; "tecle o valor de B 1/0□□□□□□□□":
      PAUSE 0
2080 LET b$=INKEY$
2090 IF b$ <> "1" AND b$ <> "0" THEN GOTO 2080
2100 LET b=VAL b$: PRINT AT 11,17; b: IF b=0 THEN PRINT
      AT 13,13; " H": GOTO 2120

```

```

2110 PRINT AT 13,13; " J"
2120 SOUND .2,15: SOUND .3,16
2130 LET s=(a OR b) AND (NOT (a AND b)): PRINT AT 10,27;s
2140 IF s=0 THEN PRINT INK 1; AT 12,10; " G": GOTO 2160
2150 PRINT INK 5; AT 12,10; " G"
2160 PAUSE 3: GOTO 1990
2170 REM *** Com este trecho se divide a tela para a
      apresentação ***
2180 PLOT 127,128: DRAW 0,-80
2190 PRINT AT 5,5; "esquema"; AT 5,19; "símbolo"
2200 RETURN

2210 REM ** Com esta parte se limpa a tela **
2220 LET a$="□□□□□□□□□□□□□□□□□□□□□□□"
2230 FOR n=4 TO 18: PRINT AT n,2;a$: SOUND .072, n*2: NEXT n
2240 PRINT AT 21,2; a$+"□": RETURN

```

O.I.- 1) Não esquema que para obter os símbolos gráficos indicados na listagem acima você deve entrar em modo gráfico com as seguintes letras:

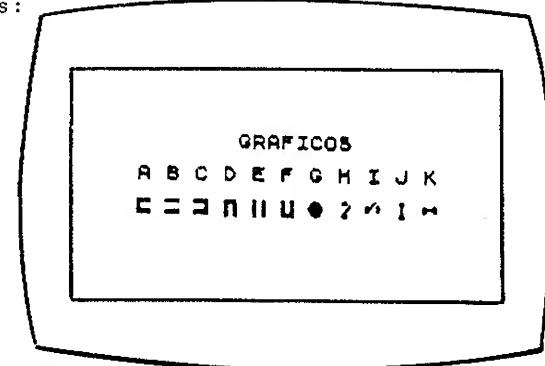


Figura 30

2) Não esquema evidentemente de gravar este super programa quando ele estiver totalmente correto ou pelo menos quando você

já tiver obtido os resultados indicados nas Figuras 31 a,b,c.

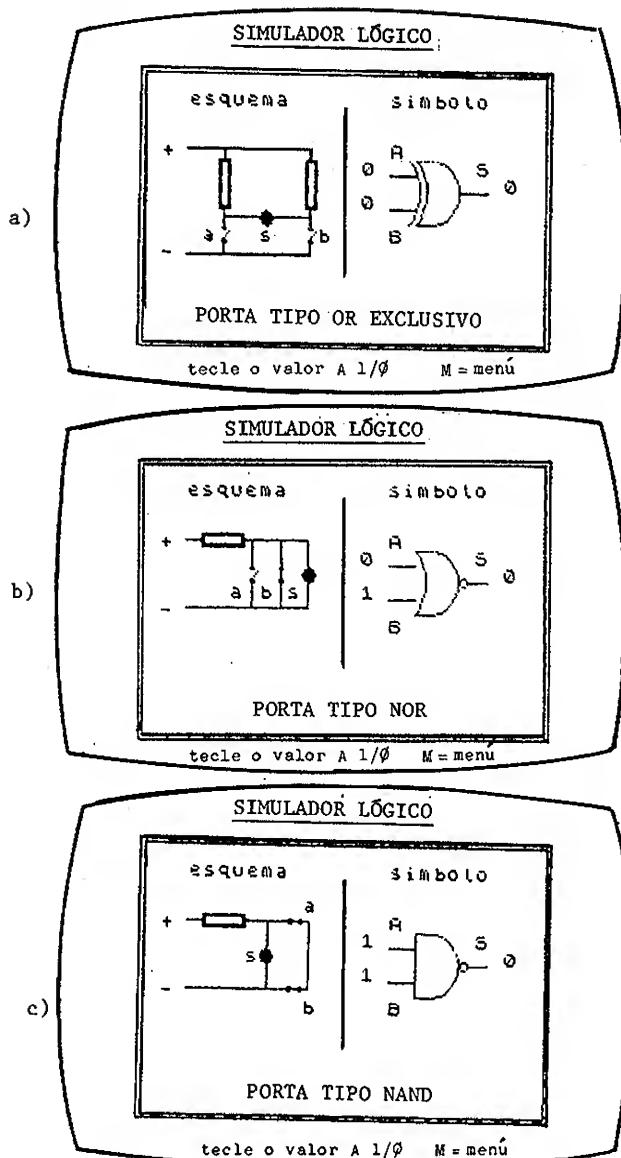


Figura 31

Para finalizar este livro (não fique triste pois já existem ou vão aparecer outros livros sobre o TK 90X do mesmo autor...) aí vai a versão computadorizada do popular jogo no qual se exige que você desloque peças em um "grid" de 4x4 até que obtenha uma sequência correta ou então um quadro certo como mostrado na Figura 32.

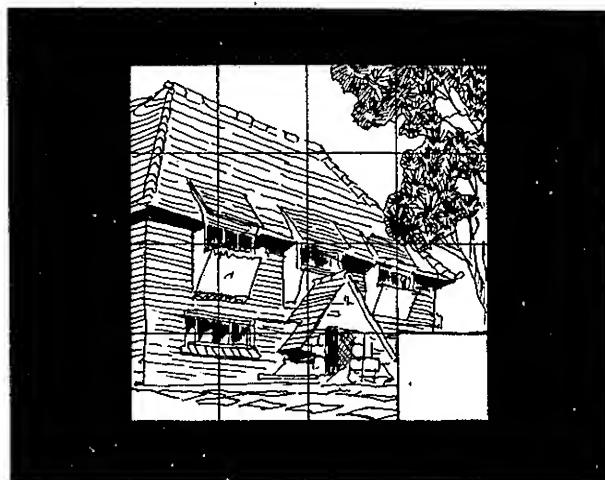


Figura 32

Na versão aqui apresentada surgirá uma casa.

O programa está pronto para você teclar e em seguida usar sem evidentemente esquecer de gravar o mesmo depois de ter se assegurado que o mesmo está correto.

Ao executar o programa você verá algumas instruções na tela e o "reticulado" no qual está a solução correta.

Depois de ter visto isto a solução é substituída pela mensagem "CRIANDO O PROBLEMA".

O tempo que o seu TK 90X irá levar para este fim é de aproximadamente 1 minuto e por isto não fique afliito(a) durante este intervalo de tempo...

Quando o problema já está criado o "grid" é exibido com as peças do quebra-cabeça colocadas em locais errados e a sua tarefa é colocá-las nos lugares certos no menor número possível de movimentos.

Para deslocar uma peça para uma casela vazia você deve teclar a letra correspondente ao quadrado onde está a peça em questão, isto é, as letras adjacentes na horizontal ou na vertical.

Os quadrados no reticulado do quebra-cabeça são representados pelas letras A até P como está indicado na Tabela 2.

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Tabela 2

O problema ou seja o quebra-cabeça está resolvido quando o quadro correto aparecer no "grid" (só faltava não ser assim, não é?).

- 0.I.
- 1) O programa deve ser embutido no TK 90X utilizando o **CAPS LOCK** (só letras maiúsculas).
  - 2) Fique atento para entrar com o número correto de espaços em branco.



Bem, aí vai o último programa (não pare para se lamentar, pois tem mais livros nossos sobre o assunto...).



```

2 POKE 23658,8
5 DIM P(16,9): DIM B(16)
10 BORDER 1: INK 0: LET T=0
15 PRINT AT 0,3; "□";:FLASH 1: PRINT "O QUEBRA-CABEÇA
DESLIZANTE";:FLASH 1
20 PRINT: PRINT
24 PRINT " A IDÉIA DESTE JOGO É VOCÊ RECREAR UM QUADRO EM
UM RETICULADO 4x4 DESLOCANDO AS PEÇAS PARA A CASELA
VAZIA."
26 PRINT
28 PRINT "VOCÊ PODE CONSEGUIR ISTO TECLANDO A LETRA DA PEÇA
QUE QUER DESLOCAR PARA O QUADRADO VAZIO."
30 PRINT: PRINT "AS PEÇAS NO QUADRÍCULADO SÃO REPRESENTADAS
PELAS LETRAS;"
35 PRINT
40 FOR J=1 TO 16
45 IF J < 5 THEN PRINT AT 15,12+J; CHR$ (64+J): GOTO 65
50 IF J < 9 THEN PRINT AT 16,8+J; CHR$ (64+J): GOTO 65
55 IF J < 13 THEN PRINT AT 17,4+J; CHR$ (64+J): GOTO 65
60 PRINT AT 18,J; CHR$ (64+J): GOTO 65
65 NEXT J
70 PRINT AT 20,2; "□"; FLASH 1: PRINT "APERTE QUALQUER
TECLA";: FLASH 0
75 IF INKEY$ = "" THEN GOTO 75
80 CLS
85 PAPER 5: GOSUB 1000: GOSUB 1300: GOSUB 1100
90 PRINT AT 4,1; "ESTA É"
95 PRINT AT 5,3; "A"
100 PRINT AT 6,1; "SOLUÇÃO"
110 FOR P=1 TO 16
112 LET B(P) = P
115 LET R=P
116 IF R=BRANCO THEN LET LUGAR=P: GOTO 125

```

```

120 GOSUB 1200
125 NEXT P
130 FOR Z=1 TO 1499; NEXT Z
135 PAPER 5: CLS
140 GOSUB 1400
145 GOSUB 1300: GOSUB 1100
150 FOR P=1 TO 16
155 LET R=B(P)
160 IF R=BRANCO THEN LET LUGAR=P:GOTO 170
165 GOSUB 1200
170 NEXT P
175 INK Ø: PAPER 5: PRINT AT 5,1;"NÚMERO";: PRINT AT 6,1;
  "DE PASSOS": PRINT AT 8,6;"Ø"
180 LET B$=INKEY$: IF B$="" THEN GOTO 180
185 IF CODE B$ < 65 OR CODE B$ > 80 THEN GOTO 180
190 GOSUB 1500: IF L=Ø THEN GOTO 180
195 LET P=CODE (B$)-64: GOSUB 1600
200 LET P=LUGAR: LET R=B(CODE (B$)-64): GOSUB 1200
205 LET S=LUGAR: LET LUGAR=CODE (B$)-64: LET Z=B(LUGAR):
  LET B(LUGAR)=B(S): LET B(S)=Z
210 LET T=T+1
215 INK Ø: PAPER 5
220 IF T < 10 THEN PRINT AT 8,6;T: GOTO 250
225 IF T < 100 THEN PRINT AT 8,5;T: GOTO 250
230 IF T < 1000 THEN PRINT AT 8,4;T: GOTO 250
235 STOP
250 GOSUB 1700
260 GOTO 180
1000 FOR J=1 TO 16
1005 PRINT "██████████████████████████████████████████████████████"
1010 FOR I=1 TO 9
1020 READ P(J,I)
1025 IF P(J,I)=Ø THEN LET BRANCO=J: GOTO 1040

```



```

1030 NEXT I
1040 NEXT J
1050 FOR K=1 TO 6: PRINT "████████████████████████████████████"
  PAUSE 6: NEXT K
1100 INK Ø
1105 FOR J=Ø TO 16
1110 PRINT AT 4,1Ø+J;CHR$ 143
1115 PRINT AT 8,1Ø+J;CHR$ 143
1120 PRINT AT 12,1Ø+J;CHR$ 143
1125 PRINT AT 16,1Ø+J;CHR$ 143
1130 PRINT AT 2Ø,1Ø+J;CHR$ 143
1135 PRINT AT 4+J,1Ø;CHR$ 143
1140 PRINT AT 4+J,14;CHR$ 143
1145 PRINT AT 4+J,18;CHR$ 143
1150 PRINT AT 4+J,22;CHR$ 143
1155 PRINT AT 4+J,26;CHR$ 143
1160 NEXT J
1165 RETURN
1200 LET X=5+4*INT ((P-1)/4)
1205 LET Y=7+4*(P-4*INT((P-1)/4))
1210 INK 4: PAPER 6
1215 PRINT AT X,Y;CHR$ P(R,1)
1220 PRINT AT X,Y+1;CHR$ P(R,2)
1225 PRINT AT X,Y+2;CHR$ P(R,3)
1230 PRINT AT X+1,Y;CHR$ P(R,4)
1235 PRINT AT X+1,Y+1;CHR$ P(R,5)
1240 PRINT AT X+1,Y+2;CHR$ P(R,6)
1245 PRINT AT X+2,Y;CHR$ P(R,7)
1250 PRINT AT X+2,Y+1;CHR$ P(R,8)
1255 PRINT AT X+2,Y+2;CHR$ P(R,9)
1260 RETURN
1300 INK Ø: PRINT AT 1,3; "□";: FLASH 1: PRINT "O QUEBRA-
  CABEÇA DESLIZANTE";: FLASH Ø
1310 PRINT AT 2,7; "ESTE É O QUADRUCULADO"

```



```

1320 FOR J=1 TO 16
1325 IF J < 5 THEN PRINT AT 17,2+J;CHR$ (64+J): GOTO 1340
1330 IF J < 9 THEN PRINT AT 18,J-2;CHR$ (64+J): GOTO 1340
1334 IF J < 13 THEN PRINT AT 19,J-6;CHR$ (64+J): GOTO 1340
1337 PRINT AT 20,J-10;CHR$ (64+J): GOTO 1340
1340 NEXT J
1350 RETURN
1400 INK Ø: PRINT AT 10,7; "CRIANDO O PROBLEMA"
1405 FOR J=1 TO 56
1410 LET B=1+INT (16*RND)
1420 LET B$=CHR$ (B+64): GOSUB 1500
1425 IF L=Ø THEN GOTO 1410
1430 IF J < 25 THEN PRINT AT 8,3+J;"*": GOTO 1450
1435 IF J < 29 THEN PRINT AT 3-16,27;"*": GOTO 1450
1440 IF J < 52 THEN PRINT AT 12,56-J;"*": GOTO 1450
1445 PRINT AT 64-J,4;"*"
1450 LET S=LUGAR: LET LUGAR=B: LET Z=B(LUGAR):
    LET B(LUGAR)=B(S): LET B(S)=Z
1455 NEXT J
1460 CLS: RETURN
1500 LET L=Ø
1501 IF CODE (B$)-64=LUGAR THEN RETURN
1505 IF LUGAR=1 AND (B$="B" OR B$="E") THEN LET L=1: RETURN
1510 IF LUGAR=2 AND (B$="A" OR B$="C" OR B$="F")
    THEN LET L=1: RETURN
1515 IF LUGAR=3 AND (B$="B" OR B$="D" OR B$="G") THEN LET
    L=1: RETURN
1520 IF LUGAR=4 AND (B$="C" OR B$="H") THEN LET L=1: RETURN
1525 IF LUGAR=5 AND (B$="A" OR B$="F" OR B$="I") THEN LET
    L=1: RETURN
1530 IF LUGAR=6 AND (B$="B" OR B$="E" OR B$="G" OR B$="J")
    THEN LET L=1: RETURN
1535 IF LUGAR=7 AND (B$="C" OR B$="F" OR B$="H" OR
    B$="K") THEN LET L=1: RETURN

```



```

1540 IF LUGAR=8 AND (B$="D" OR B$="G" OR B$="L") THEN LET
    L=1: RETURN
1545 IF LUGAR=9 AND (B$="E" OR B$="J" OR B$="M") THEN LET
    L=1: RETURN
1550 IF LUGAR=10 AND (B$="F" OR B$="I" OR B$="K" OR
    B$="N") THEN LET L=1: RETURN
1555 IF LUGAR=11 AND (B$="G" OR B$="J" OR B$="L" OR
    B$="O") THEN LET L=1: RETURN
1560 IF LUGAR=12 AND (B$="H" OR B$="K" OR B$="P") THEN LET
    L=1: RETURN
1565 IF LUGAR=13 AND (B$="I" OR B$="N") THEN LET L=1:
    RETURN
1570 IF LUGAR=14 AND (B$="J" OR B$="M" OR B$="O") THEN LET
    L=1: RETURN
1575 IF LUGAR=15 AND (B$="K" OR B$="N" OR B$="P") THEN LET
    L=1: RETURN
1580 IF LUGAR=16 AND (B$="L" OR B$="O") THEN LET L=1:
    RETURN
1590 RETURN
1600 LET X=5+4*INT ((P-1)/4)
1610 LET Y=7+4*(P-4*INT ((P-1)/4))
1620 PAPER 5
1625 PRINT AT X,Y;"□□□"
1630 PRINT AT X+1,Y;"□□□"
1635 PRINT AT X+2,Y;"□□□"
1640 PAPER 5
1650 RETURN
1700 FOR J=1 TO 16
1710 IF B(J) <> J THEN RETURN
1720 NEXT J
1730 PAPER 6: INK 2: PRINT AT 12,2; ""; FLASH 1: PRINT
    "RESOLVIDO"; FLASH Ø
1740 STOP

```



```

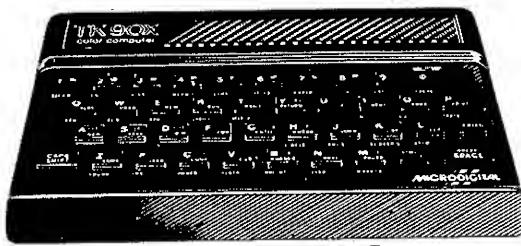
2000 DATA 0
2005 DATA 128,128,128,128,135,128,128,133,128
2010 DATA 128,128,128,128,128,137,128,128,143
2015 DATA 137,128,128,128,128,128,128,128,128
2020 DATA 128,128,141,128,141,143,128,143,143
2025 DATA 143,143,143,143,143,143,128,128,143
2030 DATA 143,143,143,143,143,143,128,128
2035 DATA 142,128,128,143,142,128,143,143,128
2040 DATA 128,143,143,128,143,143,128,143,128
2045 DATA 128,128,143,143,143,143,128,143,128
2050 DATA 143,128,128,143,143,143,128,143,128
2055 DATA 143,143,128,143,143,128,128,143,128
2060 DATA 128,143,128,128,143,143,128,128,128
2065 DATA 128,143,128,143,143,140,128,128,128
2070 DATA 128,143,128,140,143,143,128,128,128
2075 DATA 128,143,128,143,143,128,128,128,128

```

#### OUTRAS OBRAS DO AUTOR

Caderno de elementos de computação (Linguagem FORTRAN)  
 Linguagem BASIC  
 BASIC sem segredos  
 TK-divertindo  
 TK-lembrando  
 TK-calculando  
 Conhecendo e utilizando o TK 2000  
 TK-2000 na matemática  
 Dê um APPLE à sua vida  
 Imprimindo maravilhas com a GRAFIX  
 Brincando com o TRS-Color  
 BASIC no TK 90X  
 Conexão: GRAFIX-IBM PC e compatíveis  
 Gráficos no TK 2000  
 Jogos e Desenhos no TK 2000

Agora execute e tenha a sua última  
 diversão (deste livro...)



**TK 90X, o micro  
 cheio de programas.**



*Impresso nas oficinas da*  
EDITORAS PARMA LTDA.  
Fone: 209-5077  
Av. Antônio Bardella, 280  
Guarulhos - São Paulo - Brasil  
Com filmes fornecidos pelo Editor